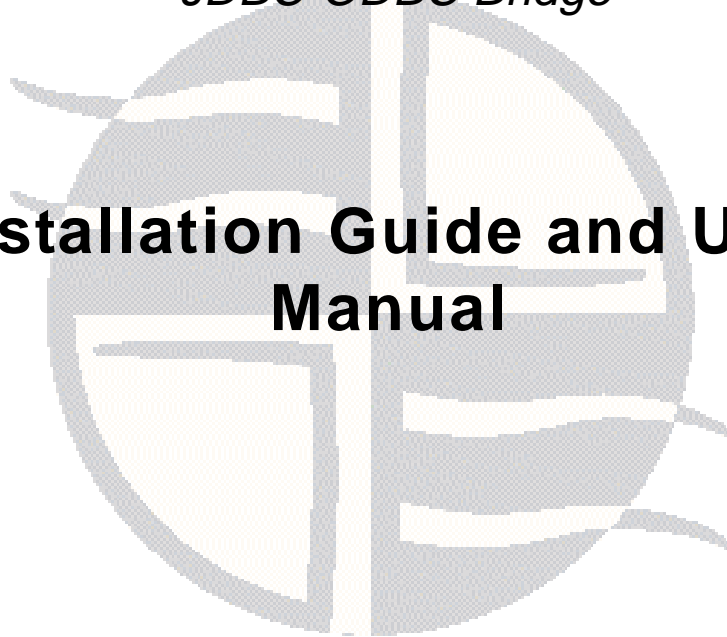


# Easysoft Data Access

*JDBC-ODBC Bridge*

## **Installation Guide and User Manual**





Version 11.

This manual documents version 1.1.n of the Easysoft JDBC-ODBC Bridge.

Publisher: Easysoft Limited

Thorp Arch Grange

Thorp Arch

Wetherby

LS23 7BA

United Kingdom

Copyright © 1993-2003 by Easysoft Limited.

All rights reserved.

You may not reverse engineer, decompile or disassemble this manual. Information in this manual is subject to change without notice. Companies, names, and data used in examples are fictitious unless otherwise noted.

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 the [Easysoft License Agreement](#)).

# CONTENTS

<b>List of Figures</b>	.....	<b>9</b>
<b>Preface</b>	.....	<b>11</b>
	Intended Audience	12
	Displaying the Manual	12
	Notational Conventions	13
	Typographical Conventions	14
	Contents	15
	Trademarks	16
<b>Chapter 1</b>	<b>Introduction</b> .....	<b>17</b>
	Why ODBC?	18
	Why JDBC?	19
	JDBC driver types	20
	Architecture	24
	Driver Managers	26
	Components	28
	Deployment	29
<b>Chapter 2</b>	<b>Installation</b> .....	<b>33</b>
	Obtaining the Easysoft JDBC-ODBC Bridge	34
	What to install	35
	Installing on Windows	38
	Uninstalling on Windows	53
	Installing on Unix	55
	Uninstalling on Unix	70

# TABLE OF CONTENTS

*Easysoft JDBC-ODBC Bridge*

<b>Chapter 3</b>	<b>Configuring the Server . . . . .</b>	<b>71</b>
	Setting up the JOB Server on Windows . . . . .	72
	Setting up the JOB Server on Unix . . . . .	79
	The Web Administrator . . . . .	87
	The Windows Registry entry . . . . .	106
	The Unix initialization file . . . . .	107
<b>Chapter 4</b>	<b>Configuring the Client . . . . .</b>	<b>109</b>
	Connecting from Java . . . . .	110
	Tuning . . . . .	120
<b>Chapter 5</b>	<b>Programming Guide . . . . .</b>	<b>121</b>
	Introduction . . . . .	122
	Java applications . . . . .	122
	Java applets . . . . .	123
	Making applets portable . . . . .	124
	Loading JOB Driver classes . . . . .	125
	Connecting to the JOB Driver . . . . .	125
	Easysoft JDBC URL attributes . . . . .	127
	Additional notes . . . . .	128
	Easysoft JOB URL examples . . . . .	129
	Minimal Java applet demonstration . . . . .	130
<b>Appendix A</b>	<b>Technical Reference . . . . .</b>	<b>133</b>
	Java runtimes and browsers . . . . .	134
	Windows platforms . . . . .	134
	Unix platforms . . . . .	134
	JDBC versions supported . . . . .	134

# TABLE OF CONTENTS

*Easysoft JDBC-ODBC Bridge*

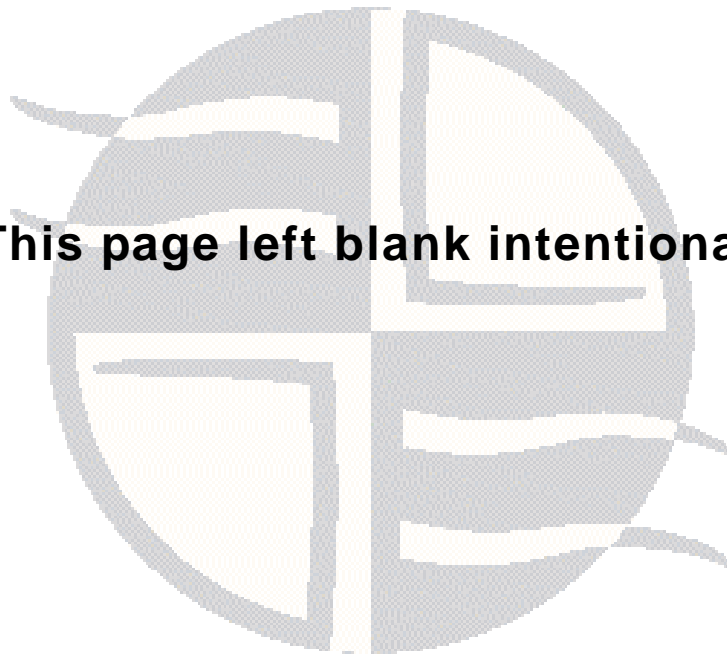
	ODBC version required .....	135
	Supported data types .....	135
	Classpath .....	135
	Tracing .....	136
	Unicode .....	138
	Firewall and security issues .....	138
	Threading .....	140
<b>Appendix B</b>	<b>JDBC Certification .....</b>	<b>143</b>
<b>Appendix C</b>	<b>Glossary .....</b>	<b>145</b>
<b>Index</b>	<b>.....</b>	<b>151</b>




# **TABLE OF CONTENTS**

*Easysoft JDBC-ODBC Bridge*

**This page left blank intentionally**





# **TABLE OF CONTENTS**

*Easysoft JDBC-ODBC Bridge*



# LIST OF FIGURES

Figure 1: Type 1 JDBC Driver architecture . . . . .	20
Figure 2: Type 2 JDBC Driver architecture . . . . .	21
Figure 3: Type 3 JDBC Driver architecture . . . . .	22
Figure 4: Type 4 JDBC Driver architecture . . . . .	23
Figure 5: Advantages of the Easysoft JDBC-ODBC Bridge . . . . .	24
Figure 6: The Driver Manager as a dynamic linker. . . . .	26
Figure 7: A Java application using JOB . . . . .	29
Figure 8: A Zero Installation Java applet using JOB . . . . .	30
Figure 9: A Zero Installation Java applet using JOB with multiple servers. . . . .	31
Figure 10: Heterogeneous database access using the ODBC Join Engine. . . . .	32
Figure 11: The Welcome dialog box . . . . .	38
Figure 12: The Software License Agreement dialog box. . . . .	39
Figure 13: The Select Components dialog box . . . . .	40
Figure 14: The Setup Type dialog box. . . . .	41
Figure 15: The User Information dialog box . . . . .	42
Figure 16: The JOB Server Administrator dialog box . . . . .	43
Figure 17: The Set Port for Internal Web Server dialog box . . . . .	44
Figure 18: The Set Port for JDBC-ODBC Bridge Server dialog box . . . . .	45
Figure 19: The Start Copying Files dialog box . . . . .	46
Figure 20: The License Manager dialog box. . . . .	47
Figure 21: The License Type dialog box . . . . .	48
Figure 22: Select the product you are licensing. . . . .	48
Figure 23: The Authorization Number dialog box . . . . .	49
Figure 24: The License Application dialog box . . . . .	49
Figure 25: The JDBC-ODBC Bridge Server Launcher dialog box. . . . .	51
Figure 26: Stopping the JOB Server . . . . .	67
Figure 27: Starting the JOB Server . . . . .	67
Figure 28: The ODBC Data Source Administrator User DSN Tab. . . . .	73
Figure 29: The ODBC Data Source Administrator System DSN Tab . . . . .	74
Figure 30: The Create New Data Source dialog box . . . . .	75
Figure 31: The Microsoft Access ODBC driver configured. . . . .	76
Figure 32: The JOB Server entry in the Services dialog box . . . . .	78
Figure 33: The Configuration dialog box for a PostgreSQL data source. . . . .	84
Figure 34: The Enter Network Password dialog box . . . . .	88

## LIST OF FIGURES

*Easysoft JDBC-ODBC Bridge*

<b>Figure 35: The Web Administrator Statistics screen . . . . .</b>	<b>89</b>
<b>Figure 36: The Web Administrator Configuration screen . . . . .</b>	<b>94</b>
<b>Figure 37: The Web Administrator Change Configuration screen . . . . .</b>	<b>101</b>
<b>Figure 38: The Web Administrator Security screen . . . . .</b>	<b>102</b>
<b>Figure 39: The Web Administrator Information screen . . . . .</b>	<b>104</b>
<b>Figure 40: The Web Administrator Client Hosts screen . . . . .</b>	<b>105</b>
<b>Figure 41: The Windows Registry entry . . . . .</b>	<b>106</b>
<b>Figure 42: The Test Applet Logon screen . . . . .</b>	<b>116</b>
<b>Figure 43: The Test Applet Connect screen . . . . .</b>	<b>117</b>
<b>Figure 44: The Test Applet JobLet screen . . . . .</b>	<b>118</b>
<b>Figure 45: The Test Applet Results screen . . . . .</b>	<b>119</b>
<b>Figure 46: An example applet tag . . . . .</b>	<b>123</b>
<b>Figure 47: An example applet tag using ez.class . . . . .</b>	<b>124</b>
<b>Figure 48: Easysoft JDBC URL attributes . . . . .</b>	<b>127</b>
<b>Figure 49: Easysoft JOB URL examples . . . . .</b>	<b>129</b>
<b>Figure 50: A minimal Java applet demonstration . . . . .</b>	<b>130</b>
<b>Figure 51: The ODBC data source selection dialog box . . . . .</b>	<b>131</b>
<b>Figure 52: Displaying the result of the tables pseudo-query . . . . .</b>	<b>131</b>

# PREFACE



---

## About this manual

This manual is intended to cover the full range of requirements for anyone wishing to install, use, or configure the Easysoft JDBC-ODBC Bridge.

Please note that this is not a JDBC programming manual.

---

## Chapter Guide

- [Intended Audience](#)
- [Displaying the Manual](#)
- [Notational Conventions](#)
- [Typographical Conventions](#)
- [Contents](#)
- [Trademarks](#)

## **PREFACE**

*About this manual*

---

### **Intended Audience**

The sections written for the Microsoft Windows platform require some familiarity with the use of buttons, menus, icons and text boxes. If you have any experience of Apple Macintosh computers, Microsoft Windows or the X Window System, you will have no difficulty with these sections.

The Unix sections require that you are experienced at using a Unix shell, and can perform basic functions like editing a file. More complex activities are detailed more clearly.

---

### **Displaying the Manual**

This manual is available in the following formats:

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

---

## Notational Conventions

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

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

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

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

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

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

**Linux**

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

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

**Caution!**

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

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

## PREFACE

*About this manual*

---

### Typographical Conventions

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

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

Click **Next** to continue.

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

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

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

At the command prompt type `admin`.

- Keyboard Commands

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

Press *<F1>* for help.

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

---

## **Contents**

- **Introduction**

Gives an overview of the ODBC and JDBC architecture, and shows what the Easysoft JDBC-ODBC Bridge brings to it.

- **Installation**

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

- **Configuring the Server**

Explains some configurable parameters in the server and the choices for deploying your Easysoft JDBC-ODBC Bridge solution.

- **Configuring the Client**

Shows you how to make your ODBC data source available for JDBC connections, and the JDBC URL you should use.

- **Programming Guide**

Information for Java developers who wish to add JDBC connectivity to their applications using the Easysoft JDBC-ODBC Bridge.

- **Appendices**

Comprising a Technical Reference, JDBC Certification document and a Glossary.

## **PREFACE**

*About this manual*

---

### **Trademarks**

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

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

Easysoft and Easysoft Data Access are trademarks of Easysoft Limited.



# INTRODUCTION

---

## Introducing the Easysoft JDBC-ODBC Bridge

The Easysoft JDBC-ODBC Bridge (JOB) is a Type 3 JDBC Driver that enables 100% pure Java programs to make calls across the network from any Java-enabled platform to an ODBC data source running on Windows or Unix.

The interface at the client end is JDBC, allowing the Easysoft JDBC-ODBC Bridge to be incorporated into ready-written Java database-aware programs.

The Easysoft JDBC-ODBC Bridge implements all the functionality required for working with Java 2, Enterprise Edition (J2EE) compatible products and is "Certified for J2EE" under the certification program managed by KeyLabs, Inc. on behalf of Sun Microsystems, Inc. (see [JDBC Certification on page 143](#)).

---

### Chapter Guide

- [Why ODBC?](#)
- [Why JDBC?](#)
- [JDBC driver types](#)
- [Architecture](#)
- [Driver Managers](#)
- [Components](#)
- [Deployment](#)

---

**Why ODBC?**

ODBC is an API (*Application Programming Interface*) definition, compliant with ANSI SQL and X/Open's SQL Call Level Interface (CLI).

Applications can be written to this API standard without considering the intricacies of the various database engines, as the ODBC *driver* takes care of all the database-specific code, if necessary mapping the structure of the underlying system to a relational framework.

ODBC permits the DBMS-specific parts of the program to be separated from the part that fulfils the business requirement.

The result is that the completed application can be attached to any Database Management System (DBMS) that has a corresponding driver.

**REF**

For the definitive SQL CLI document consult the **Open Group CAE Specification C451**, ISBN 1-85912-081-4 at <http://www.opengroup.org/pubs/catalog/c451.htm>.

The **Microsoft ODBC 3.0 Programmer's Reference** explains ODBC usage in some detail.

---

## **Why JDBC?**

Given that you are using Java and you need access to a database, JDBC is really your only option. JDBC is the standard database Application Program Interface (API) for Java, supported by Sun and many large database vendors.

JDBC is modelled on ODBC but in addition provides an object-oriented model for accessing databases, permitting use of Java methods as well as SQL for querying and updating data.

The JDBC standard means that applications can be written without considering what driver will be used in the final deployment, and gives system managers the freedom to change database engines without requiring a change in program logic.

With the Easysoft JDBC-ODBC Bridge, you can connect your Java programs to any ODBC-conformant database without waiting to see whether the relevant JDBC drivers become available.

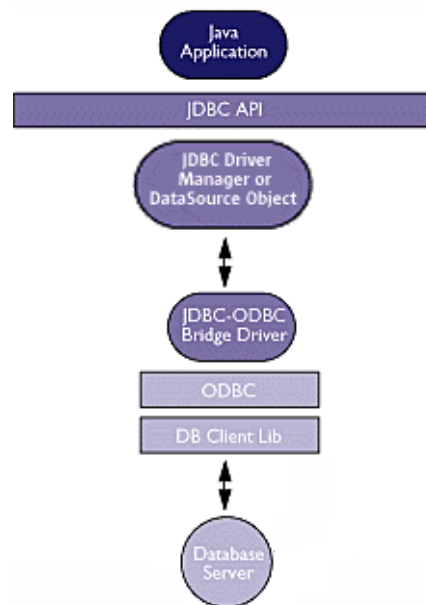
---

## JDBC driver types

The Sun website at

<http://java.sun.com/products/jdbc/datasheet.html> identifies four JDBC Driver types to help customers select the correct driver (these explanations use Sun's own terminology and diagrams):

**TYPE 1:** *A JDBC-ODBC Bridge (not the Easysoft JDBC-ODBC Bridge)*

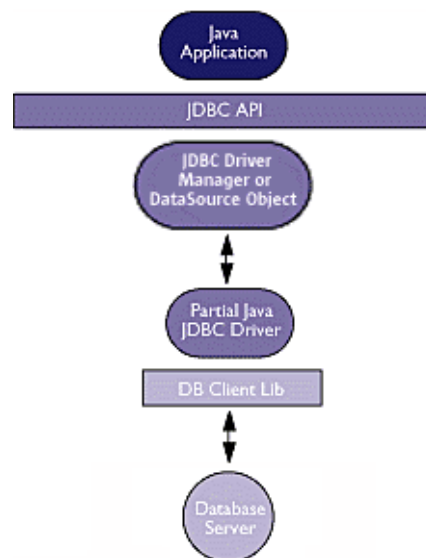


**Figure 1: Type 1 JDBC Driver architecture**

Type 1 JDBC drivers are always written in platform-specific code and thus limit the potential of the application for cross-platform deployment.

This simple JDBC driver is provided by Sun with the Java Development Kit and implements the JDBC API by making ODBC calls, allowing connection to any ODBC data source available on the local machine.

## TYPE 2: *Native-API Partly Java technology-enabled driver*



**Figure 2: Type 2 JDBC Driver architecture**

Type 2 JDBC drivers talk directly to the API of the DBMS, rather than any mapping layer such as ODBC, and can be accessed from Java.

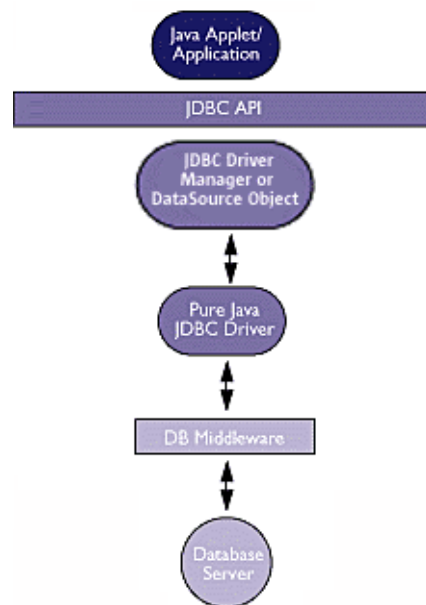
This implies that the driver either does not provide the complete JDBC API (but provides enough to drive the native database API) or is not written completely in Java (thus losing out in cross-platform functionality).

## INTRODUCTION

*Introducing the Easysoft JDBC-ODBC Bridge*

Technically, this type of driver is the most efficient user of machine resources, but this advantage is far outweighed by the need to write a different JDBC driver for each DBMS and for each platform.

**TYPE 3:** *Net-protocol fully Java technology-enabled driver*



**Figure 3: Type 3 JDBC Driver architecture**

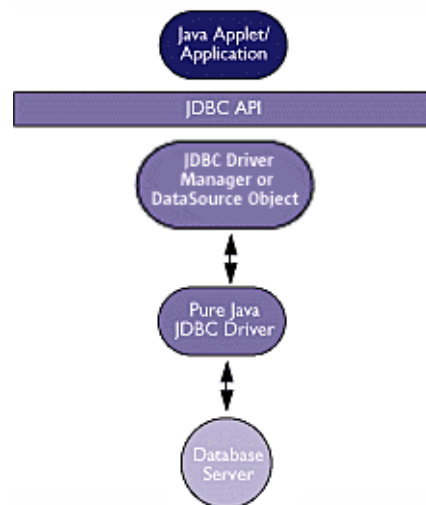
Type 3 JDBC drivers are client-server pairs which use a database-independent network protocol to bridge the network.

The client is written in Java, enabling cross-platform deployment of the application.

Sun cites this as generally "the most flexible JDBC API alternative", as the location of the database is not confined to the same machine as the Java program, and the client can be written in pure Java.

Platform and database dependent code is moved to the server running the ODBC driver, allowing these drivers to run over the internet.

**TYPE 4:** *Native-protocol fully Java technology-enabled driver*



**Figure 4: Type 4 JDBC Driver architecture**

Type 4 JDBC drivers are written entirely in Java and transmit database-specific messages direct to the DBMS, which cannot be changed without modifying or replacing the driver.

## INTRODUCTION

*Introducing the Easysoft JDBC-ODBC Bridge*

---

### Architecture

At the server end, although the Easysoft JDBC-ODBC Bridge has a Type 3 architecture, it produces ODBC calls which can be picked up by any ODBC driver and used to drive the database, rather than DBMS-specific ('native') calls.

The JOB Server is dependent only on the platform and the database engine can be replaced without users, application programmers or the Easysoft JDBC-ODBC Bridge knowing and as long as the chosen DBMS supports ODBC the code can remain unchanged.

At the application end, the driver is pure Java and therefore completely portable.

The following table summarizes the advantages of the Easysoft JDBC-ODBC Bridge:

	Type 1	Type 2	Type 3	Type 4	Easysoft
<b>Can connect to ODBC data sources</b>	Yes	No	Not always	No	Yes
<b>Application has "write once/run anywhere" platform independence</b>	No	No	Yes	Yes	Yes
<b>Can connect across internet</b>	No	No	Subject to firewall	Subject to firewall	Subject to firewall
<b>Can replace database engine without changing JDBC driver</b>	Yes	No	Not normally	No	Yes

**Figure 5: Advantages of the Easysoft JDBC-ODBC Bridge**



### **NB**

In practice, the SQL syntax and semantics vary slightly between database vendors. By carefully partitioning your code however, the database-specific functions can be stored on the server, so that the deployed programs do not need to be modified.

Note that if you want to utilize the support of the Easysoft JDBC-ODBC Bridge for JDBC 2 functionality within your applet then you must amend your HTML to work with the Java Plug-in, which users may install on their client in order to override their web browser's default virtual machine and take full advantage of the latest capabilities and features of the Java 2 SDK.

The Java Plug-in is available as a component of Sun's Java 2 Runtime Environment and you can obtain the Java Plug-in HTML Converter and supporting documentation from <http://java.sun.com/products/plugin>.

In order to develop and test for JDBC 2 functionality you need to:

- use the Java Plug-in HTML Converter to convert the applet tags on your HTML web pages into a format that can be used with the Java Plug-in.
- obtain the Java Plug-in from Sun.

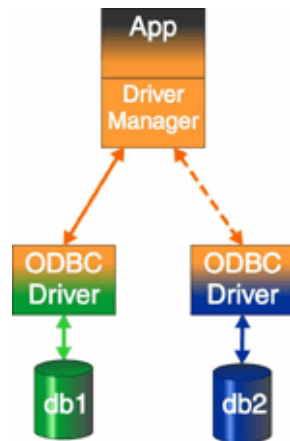
---

## Driver Managers

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

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

Instead, the application program is linked to a *driver manager*, which is responsible for loading the required ODBC driver at runtime and then initializing the driver with a stored set of attributes:



**Figure 6: The Driver Manager as a dynamic linker**

To support user-transparent replacement of the back end the driver manager provides a mapping from an abstract *Data Source Name* to a driver along with a default set of *attribute=value* pairs which are passed to the driver on connect.

## INTRODUCTION

*Introducing the Easysoft JDBC-ODBC Bridge*

Although it is possible to use ODBC without a driver manager by linking directly with the ODBC driver at build time, this option needlessly removes some beneficial functionality and the method of linking programs with the driver manager shared object (Unix) or DLL (Windows) provides three key results:

- Applications written by developers to satisfy a business requirement can be ‘plugged in’ to any compatible database.
- Administrators can connect a variety of applications (such as query tools) to their databases to browse and investigate data.
- *Data access middleware* can be inserted between the ODBC driver manager and driver to add strategic functionality such as joining heterogeneous databases into one data source or bridging a network.

On Unix, the unixODBC driver manager is included in the Easysoft JDBC-ODBC Bridge distribution.

Windows users should have the Windows ODBC driver manager included with the ODBC Driver for their database application.

### REF

For the Microsoft Windows driver manager, go to <http://support.microsoft.com> and search for MDAC, *Microsoft Data Access Components*.

See <http://www.unixodbc.org> for the full set of unixODBC data access components.

## INTRODUCTION

*Introducing the Easysoft JDBC-ODBC Bridge*

---

### Components

The Easysoft JDBC-ODBC Bridge is installed on the server (i.e. the host where the ODBC data source is installed) and comprises of:

- The Easysoft JDBC-ODBC Bridge Server  
This is a native-code program that services incoming requests for data and executes code in the ODBC driver.
- The Easysoft JDBC-ODBC Bridge Driver class library  
This allows your JDBC application to transmit these requests across the network.
- The Web Administrator  
This allows the client computer to download the Easysoft JDBC-ODBC Bridge classes at runtime and configure the JOB Server.
- The unixODBC Driver Manager (Unix distributions only)

---

## Deployment

### SCENARIO 1: JAVA APPLICATION

Scenario 1 shows a Java application accessing two remote ODBC data sources using the Easysoft JDBC-ODBC Bridge.

The Easysoft JDBC-ODBC Bridge Client is 100% Java and therefore the only requirements of the client device are a Java Virtual Machine and TCP/IP connectivity.

One database is hosted on the same platform as the Easysoft JDBC-ODBC Bridge Server and another on a separate device.

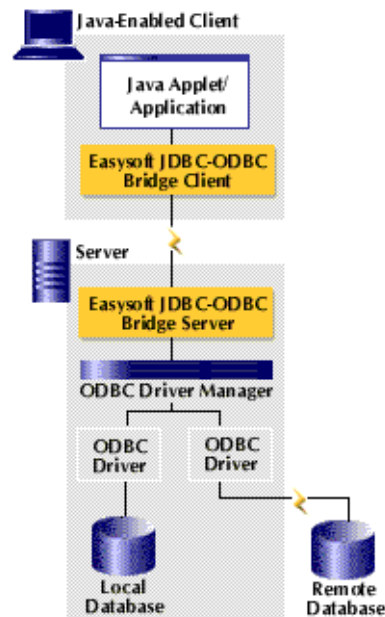


Figure 7: A Java application using JOB

## INTRODUCTION

*Introducing the Easysoft JDBC-ODBC Bridge*

### SCENARIO 2: ZERO INSTALLATION JAVA APPLET

Scenario 2 illustrates the loading of a Java applet from a Web Server.

No client installation is required.

Applet rules require that the applet and driver are served from the same machine on which the Easysoft JDBC-ODBC Bridge Server is running.

The target database can be local to the Easysoft JDBC-ODBC Bridge Server or on a remote database server.

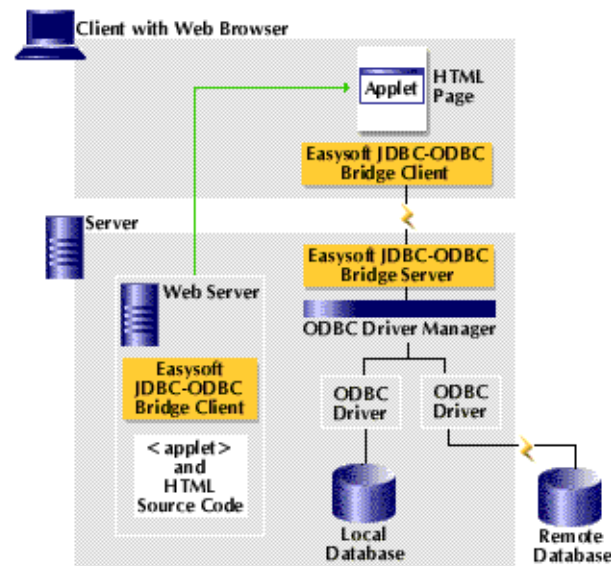


Figure 8: A Zero Installation Java applet using JOB

## SCENARIO 3: ZERO INSTALLATION JAVA APPLLET FROM A DIFFERENT MACHINE

Scenario 3 is a variation on scenario 2 and demonstrates how the HTML page containing the applet tag can be served from a different machine to the applet code and Easysoft JDBC-ODBC Bridge Client.

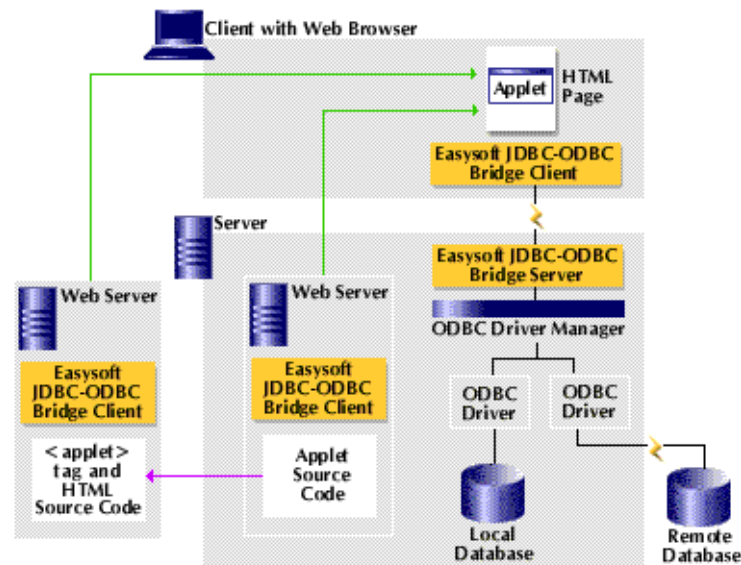


Figure 9: A Zero Installation Java applet using JOB with multiple servers

## SCENARIO 4: HETEROGENEOUS DATABASE ACCESS

Scenario 4 introduces the Easysoft ODBC Join Engine as a device to simplify and optimise multi-database queries.

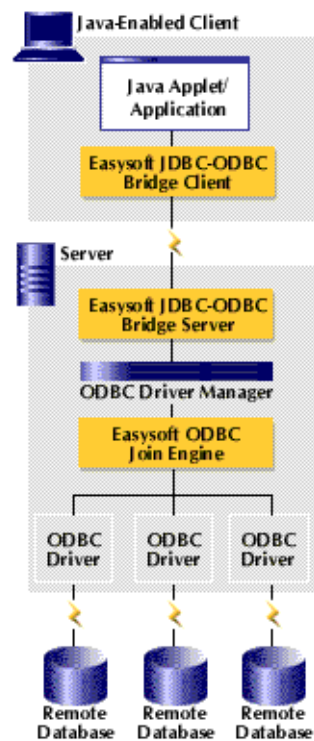
To the Java application on the client platform the three existing ODBC data sources on the Easysoft JDBC-ODBC Bridge Server appear as a single heterogeneous data source.

## INTRODUCTION

### *Introducing the Easysoft JDBC-ODBC Bridge*

Queries are passed to the Easysoft ODBC Join Engine and resolved on the Easysoft JDBC-ODBC Bridge Server platform, reducing both execution times and network traffic.

The Easysoft ODBC Join Engine can be used in this way with both Java applications and Java applets.



**Figure 10: Heterogeneous database access using the ODBC Join Engine**



# INSTALLATION

---

## Installing the Easysoft JDBC-ODBC Bridge

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

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

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

---

### Chapter Guide

- [Obtaining the Easysoft JDBC-ODBC Bridge](#)
- [What to install](#)
- [Installing on Windows](#)
- [Uninstalling on Windows](#)
- [Installing on Unix](#)
- [Uninstalling on Unix](#)

---

**Obtaining the Easysoft JDBC-ODBC Bridge**

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

- The Easysoft web site is available 24 hours a day at <http://www.easysoft.com> for downloads of definitive releases and documentation.

Select **Download** from the Easysoft JDBC-ODBC Bridge section of the website and then choose the platform release that you require.

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

- The Easysoft FTP server is available 24 hours a day at <ftp://ftp.easysoft.com>, containing free patches, upgrades, documentation and beta releases of Easysoft products, as well as definitive releases.

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

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

---

## What to install

The Easysoft JDBC-ODBC Bridge consists of the following product components:

- the JOB Driver (also known as the JOB Client)
- the JOB Server
- the unixODBC Driver Manager (on Unix platforms only)

The JOB Driver is 100% Java and can be installed on any machine supporting Java by copying it to a suitable place in the filestore and setting the appropriate value in the `CLASSPATH` environment variable.

The JOB Server is platform-specific, so be sure to download the correct distribution.

You only need to perform the download and install procedure on the server host (i.e. the machine where the ODBC data source is located).

**NB**

For the client side, copy the file `EJOB.jar` from the server host into the `CLASSPATH` on the client host. You can normally find this by pointing your browser to the URL `http://serverhost:8031/EJOB.jar` (on Windows you specify `CLASSPATH` as an environment variable via the Control Panel). If you download `EJOB.jar` from the Easysoft Web site, ensure that you save it with the `.jar` file extension and that it is NOT saved as `EJOB.html`.

In the distribution are the driver classes, the JOB Server (including the Web Administrator), shared components needed to license and run the JOB Server, and documentation files.

The Unix distribution also includes the unixODBC Driver Manager.

## INSTALLATION

*Installing the Easysoft JDBC-ODBC Bridge*

Windows users should have already received a driver manager with their ODBC Driver.

If you need the Microsoft Windows driver manager, go to <http://support.microsoft.com> and search for MDAC, **REF** *Microsoft Data Access Components*. See <http://www.unixodbc.org> for the full set of unixODBC data access components.

The name of the install file varies from platform to platform, but is of the form:

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

– OR –

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

where "x" is the major version number, "y" is the minor version number and "z" is the build index, which is incremented when minor changes are made.

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

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

within specific Unix platforms, where "*platform-variation*" refers to alternative versions available for a single platform.

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

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

### NB

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

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

### Caution!

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

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

Refer to the section relevant to your platform to continue:

- “Installing on Windows” on page 38
- “Uninstalling on Windows” on page 53
- “Installing on Unix” on page 55
- “Uninstalling on Unix” on page 70

### Win 9x

To run the Easysoft JDBC-ODBC Bridge on Windows 9x, you need WinSock2, which can be downloaded from [http://www.microsoft.com/windows95/downloads/contents/wu/admintools/s\\_wunetworkingtools/w95sockets2/](http://www.microsoft.com/windows95/downloads/contents/wu/admintools/s_wunetworkingtools/w95sockets2/).

## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

---

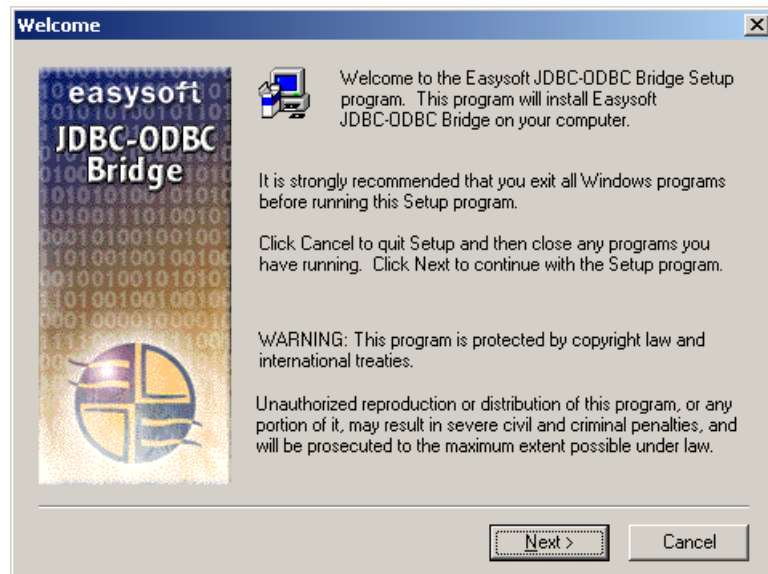
#### Installing on Windows

1. Execute the file distribution that you downloaded in “Obtaining the Easysoft JDBC-ODBC Bridge” on [page 34](#).

### Caution!

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

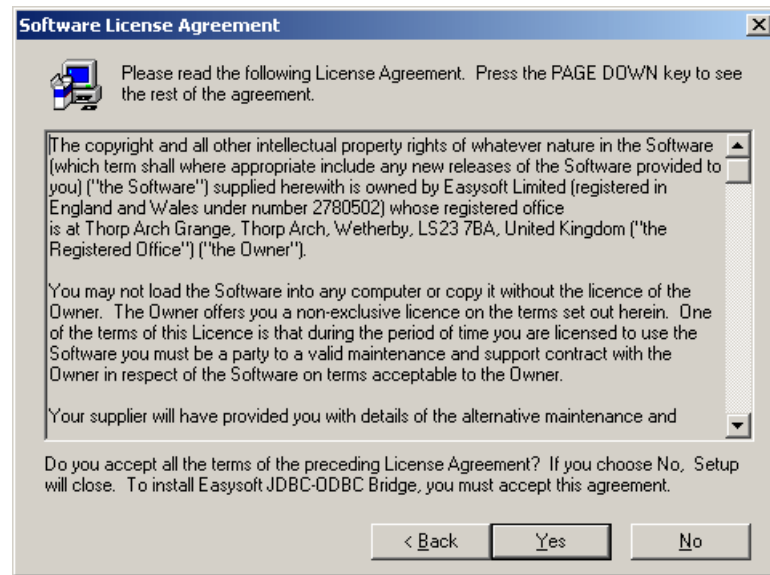
There will be a short delay while setup prepares the wizard to guide you through the rest of the install procedure before the **Welcome** dialog box is displayed:



**Figure 11: The Welcome dialog box**

2. Click **Next** to continue.

The **Software License Agreement** window is displayed:



**Figure 12: The Software License Agreement dialog box**

3. If you agree to the License Agreement click **Yes** to continue.

– OR –

If you do not agree with the license agreement click **No** and refer to the [Easysoft License Agreement](#) for more information.

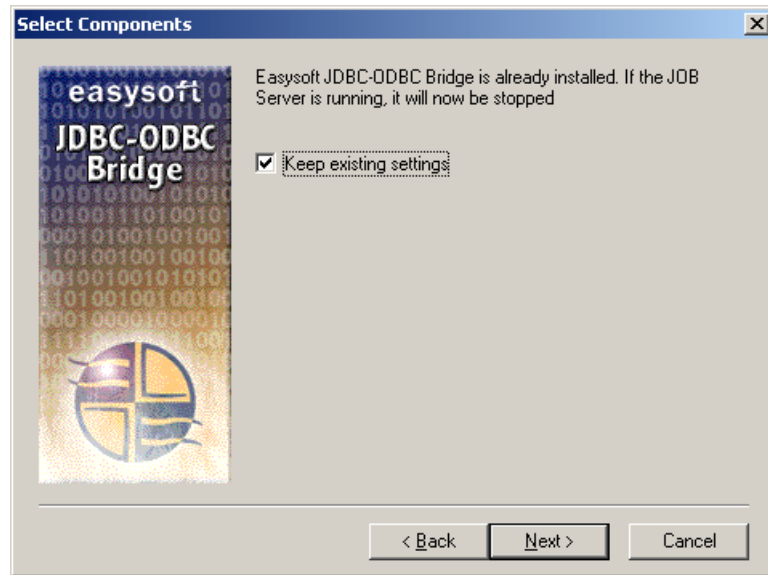
Please do not hesitate to contact Easysoft with any questions.

Easysoft contact information is given in [Contact Details](#).

## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

If there is already a version of the Easysoft JDBC-ODBC Bridge installed the **Select Components** dialog box is displayed:



**Figure 13: The Select Components dialog box**

4. Accept the default **Keep Existing Settings** value to jump to the **License Manager** screen ([step 14 on page 47](#))  
– OR –  
Unset it to continue on to [step 5 on page 40](#) and enter a new set of installation parameters.
5. Click **Next**.



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



**Figure 14: The Setup Type dialog box**

6. To change the directory into which the Easysoft JDBC-ODBC Bridge will be installed, click **Browse...** and then use the **Choose Folder** dialog box to select an alternative directory.
  7. Select **Typical** to install all components, documentation and examples, entering user information and accepting the default port numbers to be used for the Easysoft JDBC-ODBC Bridge Server and Web Administrator. This is recommended for developers and administrators.
- OR –

## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

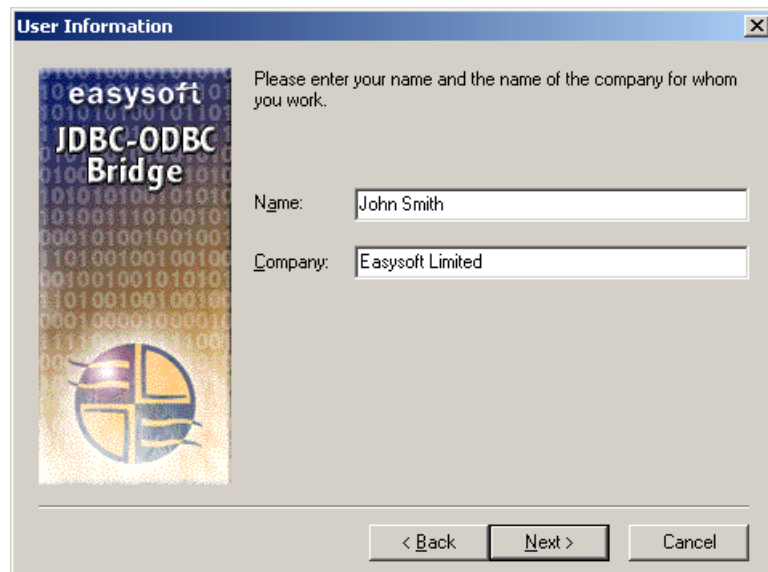
Select **Compact** to install all components, but no documentation or examples, entering user information and accepting the default port numbers to be used for the Easysoft JDBC-ODBC Bridge Server and Web Administrator. This is recommended for user machines if your administrator has already carried out a 'Typical' installation.

– OR –

Select **Custom** to install all components, but no documentation or examples, entering both user information and the port numbers to be used for the Easysoft JDBC-ODBC Bridge Server and Web Administrator. This is useful if disk space is at a premium.

8. Click **Next** to continue.

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



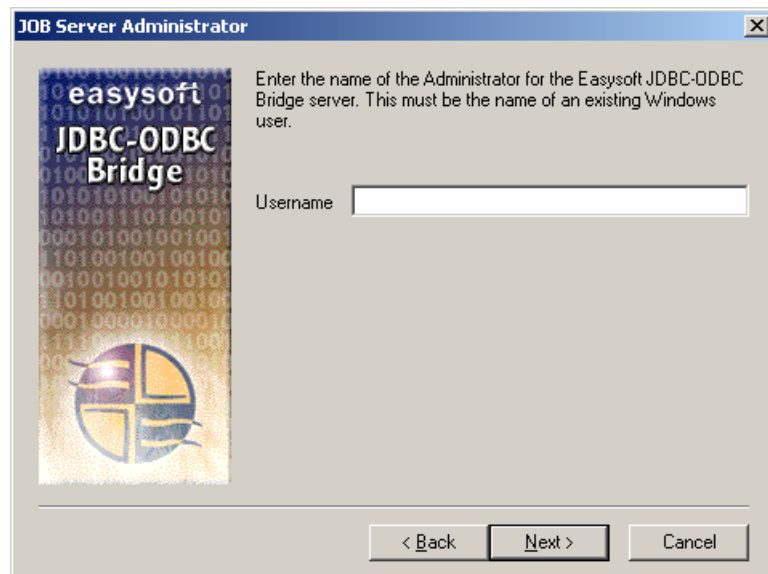
**Figure 15: The User Information dialog box**

9. Enter your name and the name of your company. Click **Next** to continue.

**NB**

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

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



**Figure 16: The JOB Server Administrator dialog box**

**Win 9x**

The **JOB Server Administrator** dialog box is not displayed under Windows 95 or Windows 98.

10. Enter a valid Windows platform user name.

This user name and associated password will be required to make any amendments to the JOB Server configuration when using the Web Administrator.

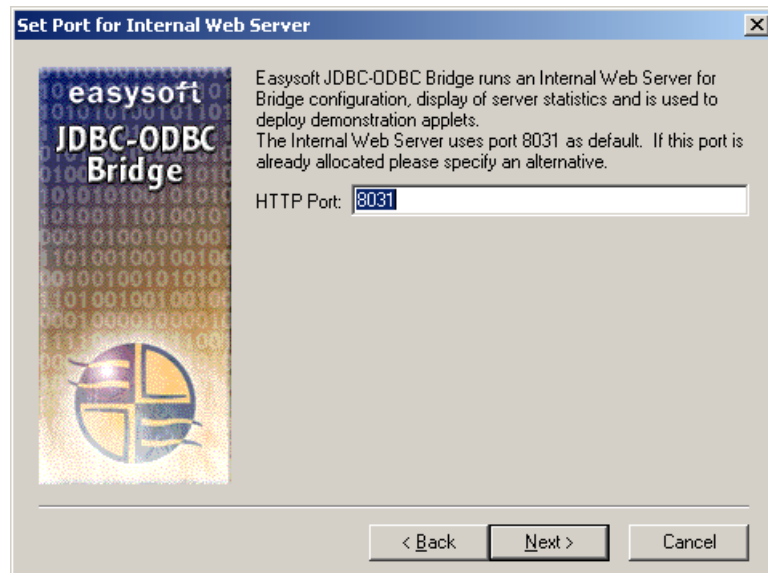
## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

If you selected **Typical** or **Compact** continue on to **step 13 on page 46**.

The user is now given the opportunity to amend the port number used by the Web Administrator (see **“The Web Administrator” on page 87**).

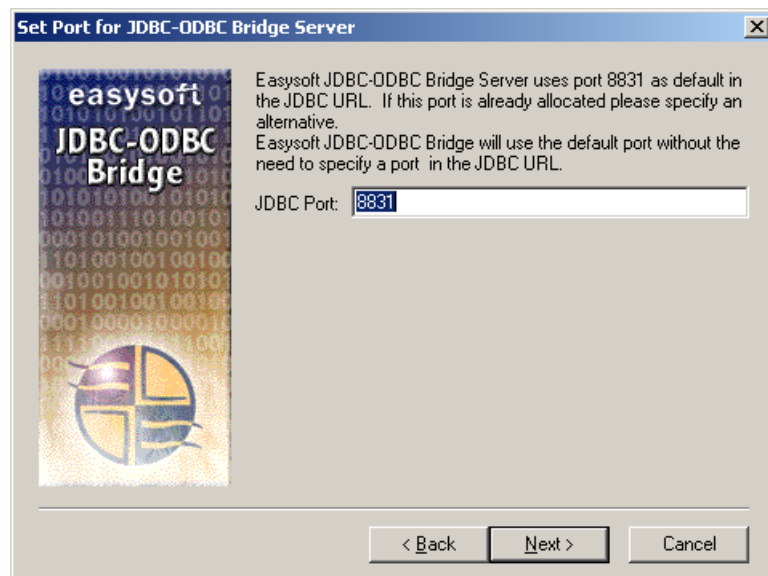
The **Set Port for Internal Web Server** dialog box is displayed:



**Figure 17: The Set Port for Internal Web Server dialog box**

11. **HTTP Port** is set to 8031 by default. Change it, if necessary, then click **Next** to continue.

The **Set Port for JDBC-ODBC Bridge Server** dialog box is displayed, allowing the port number for the JOB Server to be amended:



**Figure 18: The Set Port for JDBC-ODBC Bridge Server dialog box**

12. **JDBC Port** is set to 8831 by default. Change it, if necessary, then click **Next** to continue.

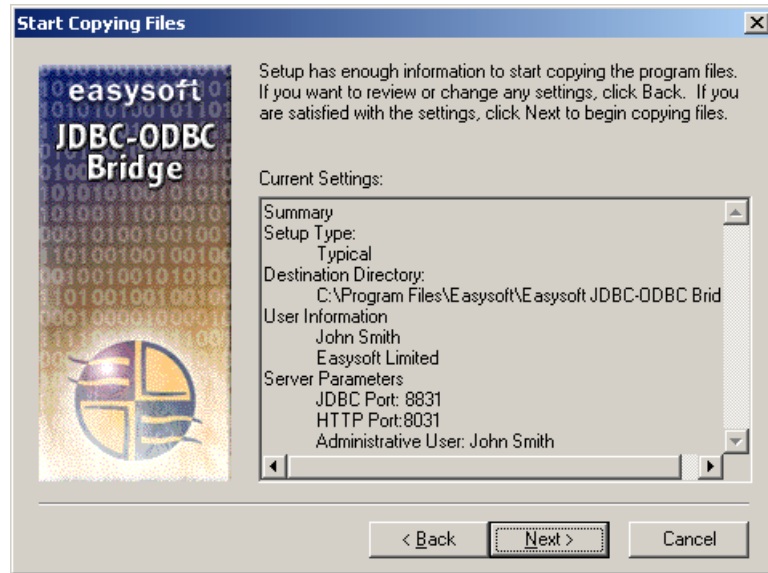
**NB**

Changing the value of **JDBC Port** will mean changing the JDBC URL used in both your application and the Test Applet.

## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

The **Start Copying Files** dialog box is displayed:



**Figure 19: The Start Copying Files dialog box**

13. Click **Next** to continue.

## LICENSING UNDER WINDOWS

The install program now starts the Easysoft License Manager (explained fully in the [Licensing Guide](#)).

You cannot use the Easysoft JDBC-ODBC Bridge until a license is obtained.

The following types of license are available:

- a **free time-limited trial license** which gives you unrestricted use of the product for a limited period (usually 14 days).

- a **full license** if you have purchased the product. On purchasing the product you are given an authorization code which you should have to hand.
14. Enter your contact details.

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

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

Easysoft Data Access 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: John Smith

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

Company: Easysoft

Telephone: 01937 860 000

Facsimile: 01937 860 001

Installed Licenses

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

Buttons: Finish, Help, Request License, Remove License, Remote License, Enter License

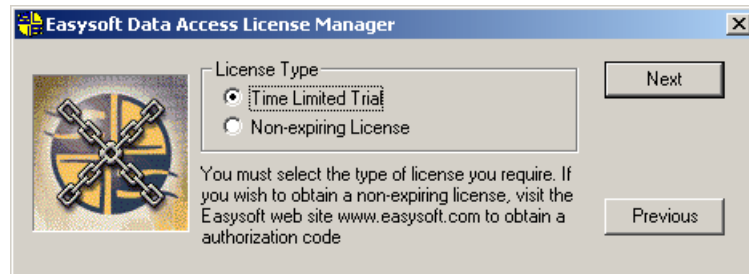
Figure 20: The License Manager dialog box

## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

15. Click **Request License**.

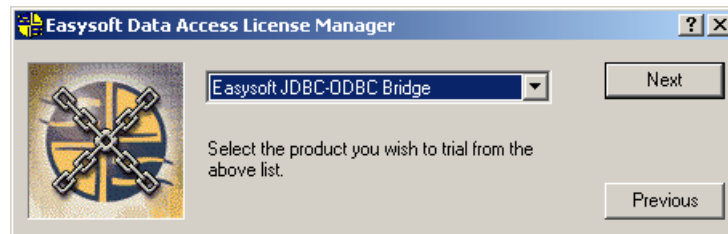
You are asked for a license type:



**Figure 21: The License Type dialog box**

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

The License Manager asks what software you are licensing:



**Figure 22: Select the product you are licensing**

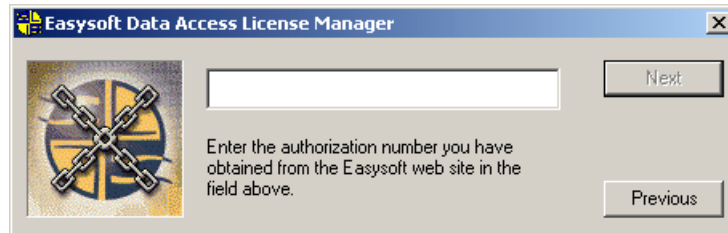
Select Easysoft JDBC-ODBC Bridge from the drop-down list and then click **Next**.

– OR –

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



The License Manager requests your authorization code:



**Figure 23: The Authorization Number dialog box**

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

The License Manager displays a summary of the details you entered and allows you to choose the method of applying for your license:



**Figure 24: The License Application dialog box**

18. Choose **On-line Request** if your machine has a connection to the internet.

## INSTALLATION

*Installing the Easysoft JDBC-ODBC Bridge*

The License Manager then transmits a network packet to the license server at Easysoft. The whole process is automatic and you can proceed to **step 19 on page 51**.

**NB**

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

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

Each of these methods involves providing Easysoft with information including your machine number (a number unique to your machine) and then waiting to receive your license key. Instead of emailing, faxing or telephoning your details to Easysoft, you can enter them directly onto the Easysoft web site and your license key will be emailed to you automatically. To use this method, click **View Request** to display your machine number, then run a web browser and go to <http://www.easysoft.com/sales/autolicense.phtml>. Choose the type of license you require, enter your machine number and then click **Continue**. Your license key will now be emailed to you. When you receive the license key, you can activate it either by double-clicking the email attachment or by clicking **Enter License** on the License Manager main screen and pasting the license key into the dialog box. A message is displayed, telling you how many licenses have been added.

**NB**

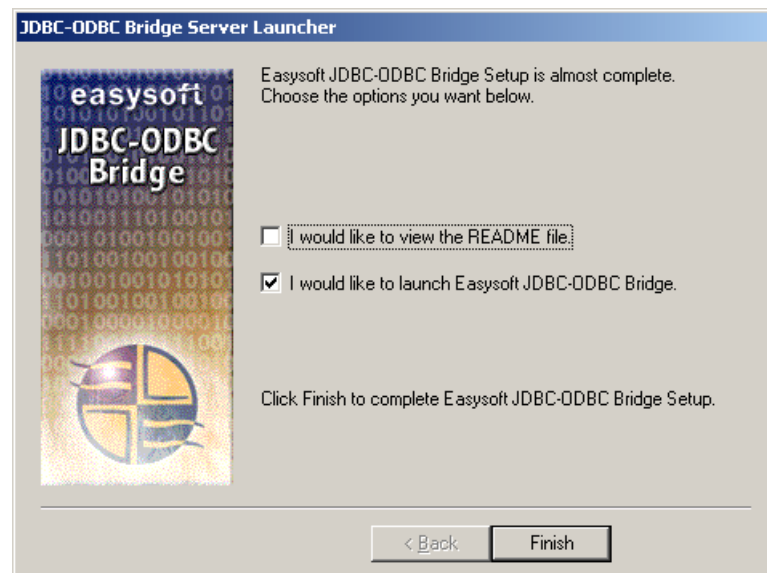
If you choose **Email Request**, the license key is emailed to the email address displayed on the License Manager main screen, not the `from:` address of your email.

For more information about the licensing procedure refer to the [Licensing Guide](#).

**NB** The JOB service must be stopped and restarted for any new licenses to take effect (see “Starting the JOB Server on Windows” on [page 77](#)).

19. Click **Finish** to close the License Manager.

The **JDBC-ODBC Bridge Server Launcher** dialog box is displayed:



**Figure 25: The JDBC-ODBC Bridge Server Launcher dialog box**

You are recommended to accept the default setting to launch the JOB Server.

**NB** The Job Server can be run manually from the start menu via **Start > Programs > Easysoft > JDBC-ODBC Bridge > Run Server**.

## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

20. Click **Finish**.

The installation process is complete.

A DOS session will start up and the Easysoft JDBC-ODBC Bridge server will launch.

The install script then opens your default browser, displaying the Web Administrator and demonstration applet.

If there is an ODBC data source already set up on your server machine, you can run the test applet to connect to it (see [“The Test Applet” on page 115](#)).

If you connect to the Internet via a proxy server you should disable the use of the proxy server for local addresses in your browser, as this will cause the connection to fail.

**NB** Disable the proxy server via **Tools > Internet Options > Connections > LAN Settings > Bypass proxy server for local addresses** under Microsoft Internet Explorer and via **Edit > Preferences > Advanced > Proxies > Direct connection to the Internet** under Netscape Navigator.

---

## Uninstalling on Windows

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

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

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

2. Select **Easysoft JDBC-ODBC Bridge** and click **Add/Remove**.
3. Click **Yes** to confirm that you wish to remove the Easysoft JDBC-ODBC Bridge and all its components.

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

**NB**

The Windows install/uninstall procedure incorporates a mechanism in the registry to determine whether or not shared files are still required by other programs.

Sometimes this database can become out-of-date, for instance if the user deleted an application directly, without using **Add/Remove Programs**, or the registry was 'repaired' after a system crash.

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

– OR –

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

## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

The uninstall process removes the Easysoft JDBC-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.

The uninstall process is complete.

---

## Installing on Unix

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

### INSTALLATION REQUIREMENTS

The Unix installation routine has the following requirements:

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

If any of these commands are missing they can be obtained from the Free Software Foundation (<http://www.fsf.org>).

- Depending on the platform, you will need up to 10Mb of free disk space for the installed programs and up to 10Mb temporary space for the installation files themselves.

### PREPARATION

1. Log on to your Unix machine as the `root` user.
2. Download the Easysoft JDBC-ODBC Bridge (see **“Obtaining the Easysoft JDBC-ODBC Bridge” on page 34**).
3. Place the distribution file in a temporary directory on your Unix machine.

## INSTALLATION

*Installing the Easysoft JDBC-ODBC Bridge*

### EXTRACTING THE INSTALLATION FILES

4. Change to the directory in which the distribution file resides.
5. Extract the installation files from the distribution file:

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

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

– OR –

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

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

– OR –

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

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

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

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

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



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

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

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

## Caution!

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

## BEGINNING THE INSTALLATION

8. Type:

```
./install
```

**NB** During the installation, you are asked to answer some questions. The default response is displayed in square brackets [], which you can press `<Enter>` to accept or you can choose any of the alternative responses shown in round brackets () by typing the required response and then pressing `<Enter>`.

Occasionally, the install program pauses to give you time to read the information displayed on screen. Press `<Enter>` to continue when you have read the current screen of information.

## INSTALLATION

*Installing the Easysoft JDBC-ODBC Bridge*

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

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

10. The script pauses to allow you to read its output so far. Up to this point it has checked:
  - that you have the minimal set of Unix programs it requires.
  - the platform you are running.
  - any platform-specific checks (such as the version of the C runtime library).
11. Press `<Enter>` to continue.

The script checks the archive package. There are three possible outcomes to this check:

- The files are checked and they pass.
  - The files are checked and some may fail.
  - 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 Easysoft JDBC-ODBC Bridge” on page 34** and download the install archive again.

– OR –

If the files passed the check then read on.

13. Press `<Enter>` to continue.

The install script then explains the situation regarding `root` privileges (i.e. if you are not the `root` user then you cannot license the software and will not be able to connect to the JOB Server until a license is applied).

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 installation directory tree.

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

If you wish to install the software in another directory, you will need to specify a directory name here.

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

**NB**

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 would accept the non-existent directory `/usr/local/odbc` and would create it, but would not accept `/usr/local/odbc/job`.

## INSTALLATION

*Installing the Easysoft JDBC-ODBC Bridge*

15. If you want a typical system-wide installation, press `<Enter>` to place the `easysoft` directory in `/usr/local/`.

– OR –

If you want a custom installation directory, type the desired directory and press `<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 Easysoft JDBC-ODBC Bridge. It asks you whether or not you wish to continue with the installation. Enter `y` or `n`.

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.

18. Press `<Enter>` to continue.

The script now extracts some files common to multiple Easysoft products.

If any of these files are already installed, the script checks whether the installed files are older and does not replace any files that are more recent than those supplied with the Easysoft JDBC-ODBC Bridge.

### INSTALLING THE SERVER

The script then asks whether you wish to install the JOB Server.

This allows you to make data sources on the local machine available to remote JDBC clients.

19. If you do not wish to install the JOB Server, enter `n` to quit the install.

– OR –

To install the server, enter `y` and read on.

20. Press `<Enter>` to continue.

The necessary files for the JOB Server are extracted.

21. Press `<Enter>` to continue.

## INSTALLING UNIXODBC AND COMMON FILES

The installation script now offers to install the unixODBC driver manager, which permits ODBC applications on the local machine to choose a data source at runtime, as the driver manager loads the required driver.

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

**NB**

unixODBC is an open source project sponsored by Easysoft and other industry members. It is rapidly becoming the standard driver manager across the Unix data access community. Easysoft recommend it for use in all Unix ODBC installations (see “**ODBC version required**” on page 135).

You should install the unixODBC driver manager if you:

- intend using multiple drivers under the JOB Server.
- already have other ODBC drivers installed on this machine.
- intend using other Easysoft products in combination with the Easysoft JDBC-ODBC Bridge.

## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

- want to use the unixODBC GUI configuration tools to define data sources.

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

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

- If there are multiple copies of unixODBC on your system you will have to choose one of them. If you pick the wrong one (i.e. not the one your applications are linked with, or not the one your run-time linker uses) then some of your ODBC drivers and data sources will not be available to the JOB Server.
- If you have built unixODBC yourself from sources, then ensure it has been configured correctly for use with your applications.

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

22. Press <Enter> to continue and then choose whether or not to install unixODBC.

If you choose to install unixODBC, its files are extracted into <InstallDir>/easysoft/unixODBC.

## LICENSING THE EASYSOFT JDBC-ODBC BRIDGE

23. The script prepares to license the product, then asks you whether you would like to run the License Manager now.

If you intend to license the software after finishing the installation, enter `n` and go to [step 32 on page 65](#).

– OR –

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

The License Manager displays a menu of options.

24. Enter the number corresponding to the Easysoft JDBC-ODBC Bridge.

The License Manager requests some contact information.

25. Enter your **Name**.

26. Enter your **Company Name**.

27. Enter at least one of **Email**, **Phone** and **Fax** (preferably all three).

28. At the **Ref** prompt, do either of the following:

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

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

If you have an internet connection you should select `1`, `Automatic`. This is the quickest and easiest method, unless your firewall or other network obstacles prevent the message getting through.

– OR –

## INSTALLATION

### *Installing the Easysoft JDBC-ODBC Bridge*

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

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

**NB**

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

If you chose the automatic licensing method and licenses are retrieved, the licensing script outputs them to a file called `licenses.out`.

If the install script detects possible new licenses in the file `licenses.out` it asks whether you want to add them to the license database.

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

– OR –

If you did not obtain licenses, enter `n`.

– OR –

If you do not see this message, then read on.

You will need to complete the offline licensing procedure as explained in **“Completing the offline licensing procedure” on page 68**.

At this point the installation will display a message to say that the Easysoft JDBC-ODBC Bridge has been successfully installed.



It will then start the JOB Server (if the JOB Server is already running then the install will ask if the user wishes to kill the existing JOB Server processes first). See **“Starting and Stopping the JOB Server” on page 66** for further details.

32. Finally, the install script starts the JOB Server and attempts to launch the Web Administrator via Netscape Navigator.

If you do not have Netscape Navigator on your server machine, run your browser and go to `localhost:8031` (or the port number you specified in **“Set Port for Internal Web Server” on page 44**) via HTTP.

The exact URL to use is displayed at the end of the install.

If you connect to the Internet via a proxy server, you should disable the use of the proxy server in your browser, as this will cause the connection to fail.

**NB** Disable the proxy server via **Tools > Internet Options > Connections > LAN Settings > Bypass proxy server for local addresses** under Microsoft Internet Explorer and via **Edit > Preferences > Advanced > Proxies > Direct connection to the Internet** under Netscape Navigator.

If there is an ODBC data source already set up on your server machine, you can run the demonstration applet via the Web Administrator to connect to it.

## INSTALLATION

*Installing the Easysoft JDBC-ODBC Bridge*

For more details about the demonstration applet, see **“The Test Applet” on page 115**.

### **NB**

If the JOB Server does not startup automatically (e.g. if it was already running before you reinstalled the software), see **“Starting and Stopping the JOB Server” on page 66** for details of starting and stopping the JOB Server manually.

### COMPLETING THE INSTALLATION

- 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 on page 60**) to be within the temporary directory, then you can safely remove the temporary directory and all its contents.

### STARTING AND STOPPING THE JOB SERVER

There may be occasions when you need to start or stop the JOB Server manually.

To do so, from the `<InstallDir>/easysoft/job` directory, type:

- `./startjob` to start the JOB Server
- `./stopjob` to stop the JOB Server

Both `./startjob` and `./stopjob` will attempt to stop all JOB Server processes:

```
Attempt to stop the JOB Server
root 13087 0.0 0.1 2968 944 pts/1 $ 11:37 0:00 ./esjobserver standalone
root 13089 0.0 0.1 2964 864 ?      $ 11:37 0:00 ./esjobserver standalone
root 13703 0.0 0.2 3312 1420 pts/1 $ 11:51 0:00 ./esjobserver standalone
yes
WARNING If you have more than two processes listed above
WARNING it indicates that there are clients currently connected
WARNING to the running job server
Do you want to kill these jobserver processes (y/n) [n]:
```

**Figure 26: Stopping the JOB Server**

There will always be one process running for the JOB Server and another for the Web Administrator in addition to any other connections.

`./startjob` will then also restart the JOB Server:

```
JOB server will start in the background
the diagnostic output of the server can be seen using the command:
tail -f /usr/local/easysoft/job/LOG
use /usr/local/easysoft/job/stopjob
to stop the JOB server.
use /usr/local/easysoft/job/startjob
to (re)start the JOB server
Point your browser to:
http://servername:8031/
To see the JOB Server Web Pages
```

**Figure 27: Starting the JOB Server**

You should consider putting these scripts into your system's startup and shutdown scripts or into your path.

**NB** To stop the JOB Server, you must be logged in as the user who started the JOB Server. This will usually be `root`.

After using `stopjob` you may need to wait for the socket to time-out before trying to start the JOB Server using `startjob`.

**Caution!**

If you run the JOB Server when logged in as any user other than `root` then server authentication is disabled, as the server will be unable to authenticate connecting JOB drivers. This is a security risk.

**COMPLETING THE OFFLINE LICENSING PROCEDURE**

If at **step 29 on page 63** you chose to write the licensing information to a file, you still need to complete the licensing procedure before the JOB Server can be activated.

Licensing information is written to the `license_request.txt` file and includes a machine number (a number unique to your machine) which Easysoft require before a license key can be issued.

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

2. When you receive your license key(s), append them to the file `/usr/local/easysoft/license/licenses`, removing any `LIC:` prefixes.

**NB**

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

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

If you need further information about licensing, please refer to the [Licensing Guide](#).

---

## Uninstalling on Unix

To remove the Easysoft JDBC-ODBC Bridge from your Unix system, you need to:

- stop the JOB Server (see “**Starting and Stopping the JOB Server**” on page 66 for details).
- remove the install directory tree.

### REMOVING THE INSTALL DIRECTORY TREE

1. Log in as the user who performed the installation (usually `root`).
2. Enter the following:

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

```
pwd
```

The system displays the current directory.

Double-check that this is the directory under which you installed the Easysoft JDBC-ODBC Bridge:

```
ls
```

*(Check that you are in the right directory).*

## Caution!

Be very careful issuing the following `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.

```
rm -r job
```

The system may ask you to confirm deletion for some files.

# CONFIGURING THE SERVER

---

## Configuring the Easysoft JDBC-ODBC Bridge Server

There are certain configurable parameters that affect the JOB Server, irrespective of the DSN to which you are connecting.

In Windows, these are stored in the registry.

In Unix they are held in the `esjobserver.ini` initialization file.

On either platform you can use the Web Administrator to modify these parameters.

---

### Chapter Guide

- [Setting up the JOB Server on Windows](#)
- [Setting up the JOB Server on Unix](#)
- [The Web Administrator](#)
- [The Windows Registry entry](#)
- [The Unix initialization file](#)

---

## Setting up the JOB Server on Windows

In Windows, data sources are made visible to applications through the Microsoft driver manager.

You set up a data source for the JOB Server in the same way as any ODBC application.

The JOB Server for Windows can connect to any *System Data Source Name* (System DSN) configured on your machine, given the necessary information.

If you do not already have a System DSN on your machine then now is the time to create one. You should use the ODBC driver suitable for your data source.

This section consists of a worked example where a System DSN is created for the Microsoft Northwind database, which is shipped with Microsoft Access.

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

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

The first step is to open the Microsoft Data Source Administrator on the machine where the JOB Server is installed:

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

**9x**

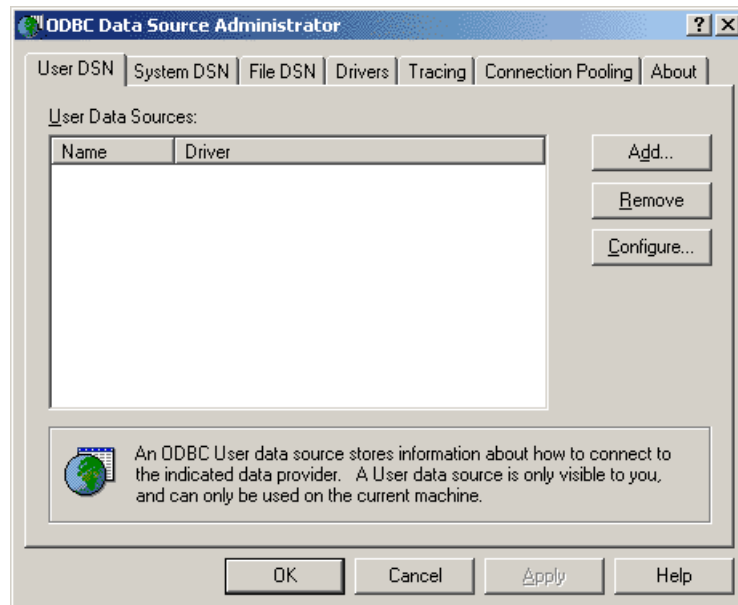
Select **Start > Settings > Control Panel** and double-click **ODBC Data Sources (32bit)**.



**NT**

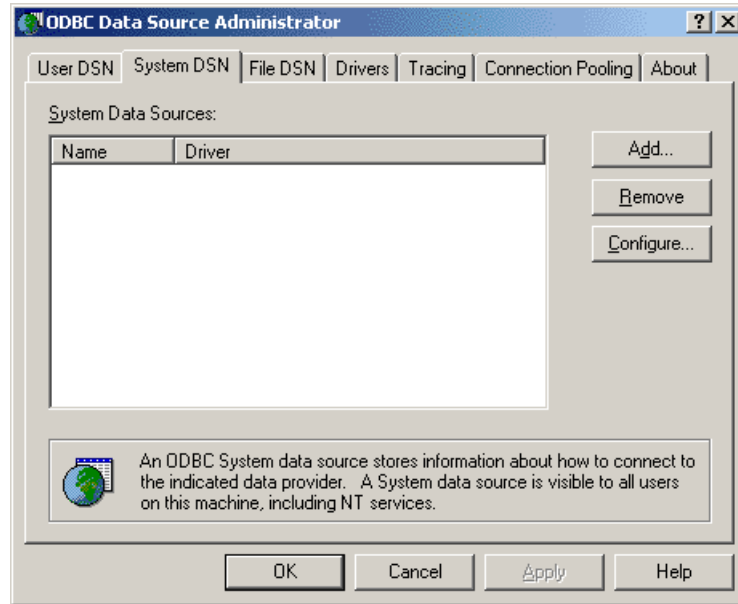
Select **Start > Settings > Control Panel** and double-click **Data Sources (ODBC)**.

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



**Figure 28: The ODBC Data Source Administrator User DSN Tab**

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



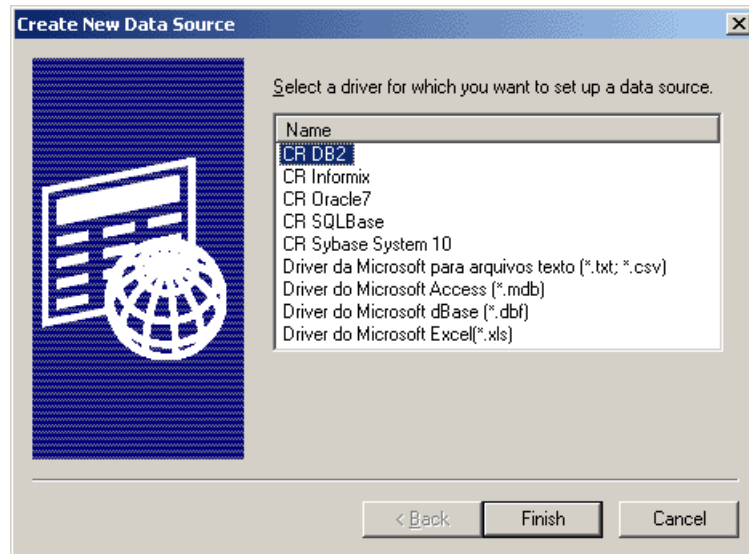
**Figure 29: The ODBC Data Source Administrator System DSN Tab**

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

Since the JOB Server runs as a service, User DSNs are not available to it.

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

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



**Figure 30: The Create New Data Source dialog box**

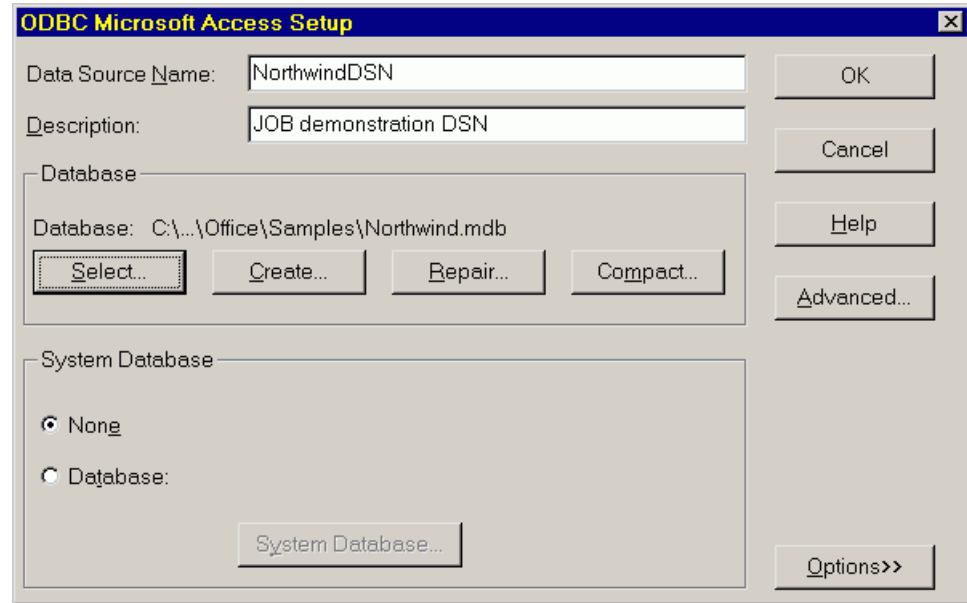
4. Select **Microsoft Access Driver** and click **Finish**.

The ODBC driver for Microsoft Access displays a dialog box for configuring the Data Source (this step differs from one database to another).

5. Enter a name for this data source in the **Data Source Name** box ("NorthwindDSN", for example).

The text that is entered into the **Description** field may be displayed to a user if they choose to connect interactively.

Enter a description, such as "JOB demonstration DSN":



**Figure 31: The Microsoft Access ODBC driver configured**

6. Click **Select...** to browse for the target database, select your chosen database and click **OK**.

This example uses 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. In this case, use any database you have to hand, preferably a small one.

**NB** Remember the Data Source Name, because it will be required when you come to connect through the Easysoft JDBC-ODBC Bridge.

7. Click **OK**.

You are returned to the ODBC Data Source Administrator window.

Note that:

- the window now contains a line with your new DSN in it
- the **System DSN** tab should be selected (if it is not, then you must remove the DSN, select the **System DSN** tab, and return to [step 3 on page 74](#)).

8. Click **OK**.

You have now set up a system-wide Data Source Name on your machine to a local database, making it visible to the JOB Server (and all ODBC programs).

### STARTING THE JOB SERVER ON WINDOWS

For your Java program to connect to the local DSN, the JOB Server must be running.

In Windows NT or 2000, the installation program configures the JOB Server to start automatically as an NT Service.

For Windows 95 or 98, the server must be run manually each time you log in to the desktop.

**95/98**

Choose **Start > Programs > Easysoft > JDBC-ODBC Bridge > JOB Server**.

Alternatively, use the command-line interface and run `server.bat`, which stops any running JOB Server and Web Administrator and then starts them both again.

For Windows 95/98, run `esjobserver` in standalone mode by typing: "`esjobserver standalone 8831`", which will start the JOB Server and display output about the connection.

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

### CHECKING THE JOB SERVER SERVICE ON WINDOWS NT/2000

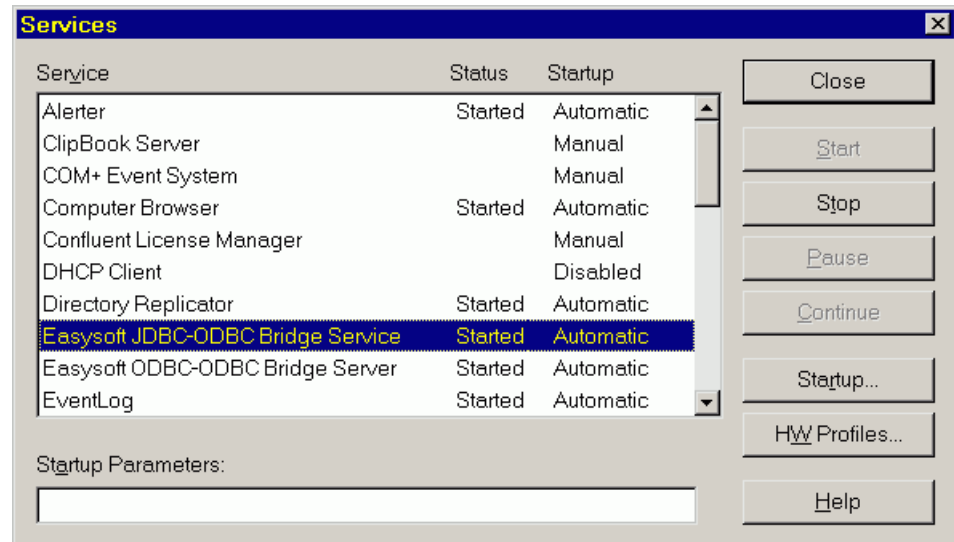
This procedure is not normally necessary, but if you are having problems connecting with the Easysoft JDBC-ODBC Bridge, it is a good idea to check through it.

Otherwise, go to "[Connecting from Java](#)" on page 110 to connect a client.

1. Choose **Start > Settings > Control Panel > Services**.

The **Services** dialog box displays all the registered services on an NT system.

Find the entry for the Easysoft JDBC-ODBC Bridge Server:



**Figure 32: The JOB Server entry in the Services dialog box**

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

You now have a JOB Server running on the Windows machine.

Refer to "**Connecting from Java**" on page 110 to find out how to connect your Java applet or application to the data source.

---

### Setting up the JOB Server on Unix

The Easysoft JDBC-ODBC Bridge is shipped as a binary executable linked with the `unixODBC` driver manager.

There are two ways to make your data source visible to the Easysoft JDBC-ODBC Bridge:

- create a System DSN on the server (see "**Using JOB with `unixODBC`**" on page 80).
- remove or rename the shared object `libodbc.so` (the suffix may differ from platform to platform) and replace it with a link to your chosen ODBC driver.

If you choose the second approach, then all Easysoft JDBC-ODBC Bridge connections will have to connect to your chosen ODBC driver.

If you use unixODBC you will be able to choose which ODBC driver to connect to at runtime.

**NB**

unixODBC is a project creating free data access components for Unix platforms. It is not an Easysoft product, but its development and maintenance are sponsored by Easysoft and other industry bodies. For further information about using unixODBC, please visit <http://www.unixodbc.org>.

**USING JOB WITH UNIXODBC**

If it is supplied with the necessary information, the JOB Server can connect to any system data source configured on a Unix machine.

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

**NB**

This section explains how to set up data sources using unixODBC, as installed by Easysoft. If you choose to use a different driver manager, you should refer to the documentation with that driver manager for details of setting up data sources on Unix.

With unixODBC, you can create a data source by either:

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



### CREATING A DSN BY EDITING A CONFIGURATION FILE

With unixODBC, data sources are stored in a configuration file called `odbc.ini`. If you accepted the default Easysoft JDBC-ODBC Bridge installation, system data sources will be stored in `/etc/odbc.ini`. If you have built unixODBC yourself, the directory is specified in the `--sysconfdir=directory` configuration option.

You usually have to be logged in as `root` to edit the system `odbc.ini`.

Each section of the `odbc.ini` file starts with a data source name in square brackets `[ ]` followed by a number of `attribute=value` pairs.

**NB** Attribute names in `odbc.ini` are not case sensitive.

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

For example, a data source using the PostgreSQL driver will be in the format:

```
[MAIN]
Description = Main data on Admin box
Driver = PostgreSQL
Trace = No
TraceFile = sql.log
Database = main
Servername = localhost
UserName =
```

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

```
Password =  
Port = 5432  
Protocol = 6.4  
ReadOnly = No  
RowVersioning = No  
ShowSystemTables = No  
ShowOidColumn = No  
FakeOidIndex = No  
ConnSettings =
```

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

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

### CREATING A DSN USING THE ODBC DATA SOURCE ADMINISTRATOR

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

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

The ODBC Data Source Administrator opens.

4. Select the **System DSN** tab.

This will create a data source which is available to any user or service that logs into this machine.

5. Click **Add** to create a new data source.

The **Adding a New Data Source** dialog box displays a list of the drivers which are available.

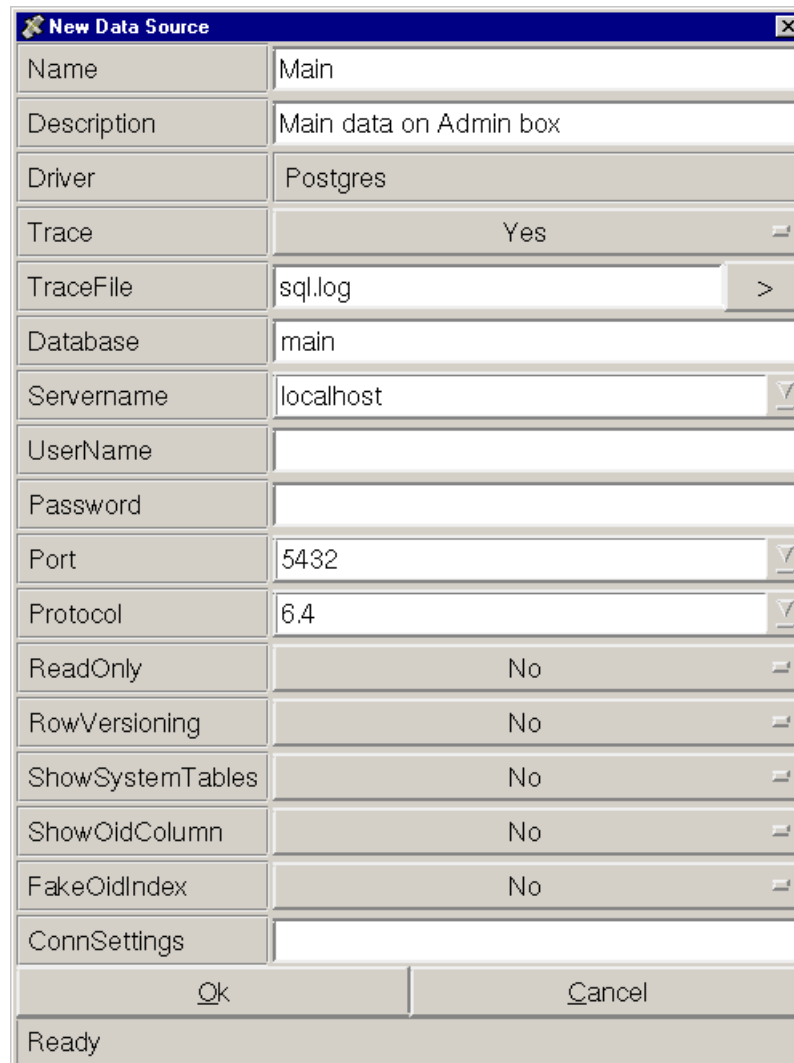
6. Select the driver that you want to use to connect to the database.
7. Click **OK**.

A driver-specific **New Data Source** dialog box is displayed.

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

The dialog box for setting up a PostgreSQL data source will be in the format:



The image shows a Windows-style dialog box titled "New Data Source". It contains a list of configuration fields for a PostgreSQL data source. The fields are as follows:

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

At the bottom of the dialog box, there are two buttons: "Ok" and "Cancel". The status bar at the very bottom of the dialog box displays the word "Ready".

**Figure 33: The Configuration dialog box for a PostgreSQL data source**

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

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

### TESTING THE DATA SOURCE

You now have a data source connecting to the database on your server.

Before attempting to connect via the Easysoft JDBC-ODBC Bridge, you should test that this data source is working so that you can be confident that the server side is functioning correctly.

You can do this using any ODBC application available on your Unix machine, or using the `isql` utility provided with unixODBC.

To use `isql` to test the data source:

1. Change into the `<InstallDir>/easysoft/unixODBC/bin` directory
2. Type:

```
./isql -v data_source_name
```

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

```
./isql -v main
```

If the DBMS on your server requires authentication, you should type:

```
./isql -v data_source_name [user_name] [user_password]
```

3. Once connected, type an SQL statement to query the data.  
e.g.

```
select * from table;
```

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

where *table* is a table in that database, or type `help` to get a list of tables in the database.

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

Once you have established that the data source on your server machine is accessing data correctly, you can establish a connection to the data source across the Easysoft JDBC-ODBC Bridge.

### STARTING THE SERVER IN UNIX

For your Java program to connect to the local DSN, the JOB Server must be running.

Normally the JOB Server will be started at the end of the installation procedure, but if you need to start or stop it manually, change into the `<InstallDir>/easysoft/job` directory and then:

- to start the JOB Server, type:

```
./startjob
```

- to stop the JOB Server, type:

```
./stopjob
```

You should consider putting these scripts into your system's startup and shutdown scripts or into your path.

After using `stopjob` you may need to wait for the socket to time-out before trying to start the JOB Server using `startjob`.

**NB**

To stop the JOB Server, you must be logged in as the user who started the JOB Server. Usually this will be `root`.

See "[Starting and Stopping the JOB Server](#)" on page 66 for more details.

---

## The Web Administrator

This section provides a tour of the Web Administrator interface and shows you how to change the configurable parameter settings in the JOB Server.

You can also edit these settings through:

- the JOB registry settings on Windows (see "[The Windows Registry entry](#)" on page 106).
- the JOB initialization file on Unix (see "[The Unix initialization file](#)" on page 107).

In order to follow this section you will need:

- a running JOB Server.
- a user name and password for the Web Administrator on Windows, as entered on the **User Information** screen during the installation of the Easysoft JDBC-ODBC Bridge (see "[Installing on Windows](#)" on page 38).

### Win

A user name and password must be entered during the installation of the Easysoft JDBC-ODBC Bridge or the Web Administrator JOB Server configuration facility will be disabled.

– OR –

the `root` user name and password on Unix.

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

### NB

The Web Administrator user name is displayed in the case sensitive HTTPAdmin field on the **Configuration** screen (see "**The Configuration Screen**" on page 94), which is hidden by default, but can be amended.

Update access will be denied if HTTPAdmin does not specify a valid user on the server system (setting it to "disabled" disallows HTTP Authentication).

Open the URL `http://serverhost:8031/` (where *serverhost* is the name of the host running the JOB Server) from a web browser.

Depending on your server configuration, you may need to specify a port address different from the Easysoft default of 8031.

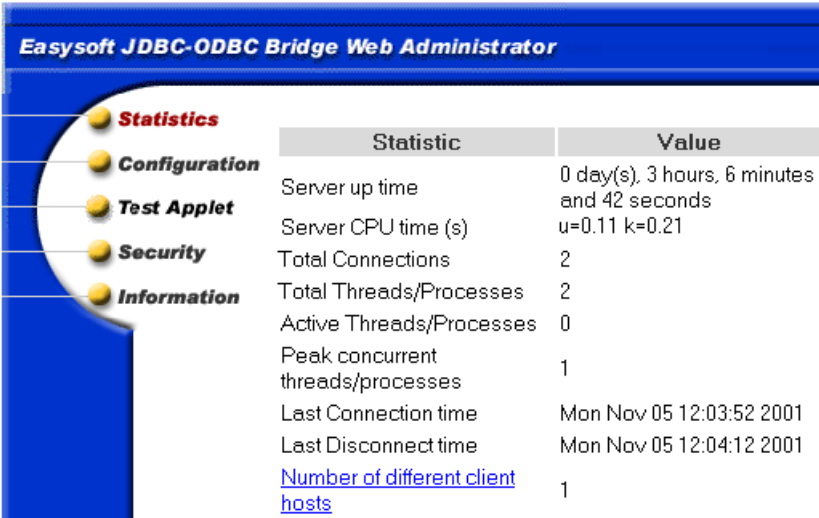
If you click on a link to a protected page you will be prompted by your browser for the Web Administrator user name and password for the ESJOBServer realm:



**Figure 34: The Enter Network Password dialog box**



The web page returned is generated by the server process.



Statistic	Value
Server up time	0 day(s), 3 hours, 6 minutes and 42 seconds
Server CPU time (s)	u=0.11 k=0.21
Total Connections	2
Total Threads/Processes	2
Active Threads/Processes	0
Peak concurrent threads/processes	1
Last Connection time	Mon Nov 05 12:03:52 2001
Last Disconnect time	Mon Nov 05 12:04:12 2001
<a href="#">Number of different client hosts</a>	1

Figure 35: The Web Administrator Statistics screen

**The Statistics Screen** displays runtime statistics for the latest run of the server and allows access to the following additional screens:

- **The Configuration Screen** allows a user to define JOB Server features.
- **The Test Applet** runs a demonstration applet, which allows the retrieval of data from any ODBC System Data Source hosted on a server.
- **The Security Screen** allows a user to define from which machines the JOB Server allows access.
- **The Information Screen** provides access to various Easysoft locations providing documentation and user support.

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

- **The Client Hosts Screen** displays details of the different clients which have connected to the JOB Server.

### THE STATISTICS SCREEN

The Web Administrator **Statistics** screen contains the following fields:

- **Server up time**

The time in days, hours, minutes and seconds since the JOB Server was started.

- **Server CPU time (s)**

This field is only visible if the `ShowProcessTime` flag is set to on (see **ShowProcessTime on page 100**).

One or more values will be shown:

- If only one value is shown it is the total CPU time consumed by the JOB Server.
- If two times are shown the first is user time and the second is kernel time.

Note that:

- when the JOB Server is running multi-threaded any of the CPU times shown include CPU time consumed by the ODBC driver manager, any ODBC drivers loaded and any child processes or threads.

- the process time is only updated when the JOB Server is idle. If the JOB Server is extremely busy servicing incoming connections then the process time will not be updated.

- **Total Connections**

The total number of connections (or attempted connections) to the JOB Server.

This will include connections dropped due to no license or insufficient license slots, port scanners or anyone using `telnet` to access the JOB Server port.

- **Total Threads/Processes**

The total number of threads or processes that the JOB Server has created during its execution.

Connections denied access because of an access control rule or `MaxThreadCount/MaxClientConnect` being exceeded are not included, because the JOB Server does not start a thread or process for these.

- **Active Threads/Processes**

The total number of active threads or processes the JOB Server has created to handle connections.

This number may exceed the actual active count as the JOB Server only looks for exited threads and processes when five seconds has elapsed without any connections (this is done to give preference to incoming connections).

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

Note that if `MaxThreadCount` or `MaxClientConnect` is set to anything other than 0 then the JOB Server has to reap exited threads and processes every time a new connection arrives.

Also note that this is not a limit and may therefore exceed your maximum licensed connection slots.

- **Peak concurrent Threads/Processes**

The highest value ever seen in the `Active Threads/Processes`.

- **Last Connection time**

The time the last connection occurred.

- **Last Disconnect time**

The time the last disconnection occurred.

- **Number of different client hosts**

The number of different client machines which have connected to the JOB Server (where a client machine is identified by its IP address).

You can click on this link to get a list of IP addresses or machine names. Machine names are only displayed if you have `ReverseLookup` enabled.

### ***Changing the refresh frequency***

The Web Administrator uses a set of template files into which the dynamic data is inserted before sending it back to your browser.

The template file for the **Statistics** screen is `index.html`, which is located in the `/admin` directory wherever you installed the JOB Server.

Edit the `index.html` file and near the top you will see:

```
meta http-equiv="refresh" content="60"; URL=/index.html
```

Change the 60 (the refresh time in seconds) to your preferred setting.

### **NB**

Note that setting the refresh time to a very low value will increase the workload on the JOB Server process which handles HTTP requests.

As this may reduce the response time to the JOB Server thread, times much less than 60 seconds are not recommended.

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

### THE CONFIGURATION SCREEN

**Easysoft JDBC-ODBC Bridge Web Administrator**

- Statistics
- Configuration**
- Test Applet
- Security
- Information

Parameter	Value
Port	8831
HTTPPort	8031
Timeout	7200
MaxThreadCount	100
RetryCount	5
RetryPause	3
SingleStatement	n
HTTPAdmin	Tim Ashford
Path	C:\Program Files\Easysoft\Easysoft JDBC-ODBC Bridge
Logging	64
LogDir	C:\temp
Authentication_Disabled	X
HTTP_Server	✓
MultiProcess	X
Reserved	X
HideSensitive	X
ReverseLookup	X
AuditODBCAccess	X
Reserved	X
ShowProcessTime	X

**Change**

Data Access by Easysoft

**Figure 36: The Web Administrator Configuration screen**

This section explains the parameters that are configurable via the Web Administrator **Configuration** screen.

These parameters apply either to the JOB Server itself, or as a default value for all DSNs accessed via the server.

Brief reminder notes are available by scrolling the browser window. Make your changes and click **Submit**. At the confirmation screen, click another menu option or wait a few seconds to return to the **Server Configuration** screen and see the changes you have made.

These settings, and some additional settings, can also be edited via the registry in Windows (see "[The Windows Registry entry](#)" on [page 106](#)) or in the `esjobserver.ini` file in Unix (see "[The Unix initialization file](#)" on [page 107](#)).

The Web Administrator **Configuration** screen displays the configurable parameters of the server process. Some sensitive settings may not be displayed.

You can modify the server settings by clicking **Change** and then typing the Web Administrator user name and password when prompted.

The Web Administrator user name is the network user entered during the installation process and the password the one required for that user on the system where the JOB Server is running.

### **NB**

You are only asked for log in details if authentication has been enabled by entering a value other than "disabled" into the HTTPAdmin field on the Web Administrator **Change Configuration** screen.

The **Configuration** screen contains the following fields (click **Submit** to make your changes).

Values are case-dependent if the operating-system is, so it is best to match case where possible:

- **Port**

The port on which the JOB Server listens for incoming JOB Client connections.

The default port number is 8831, but it may be any port number not in use on your JOB Server machine.

- **HTTPPort**

The port on which the JOB Server listens for HTTP requests (i.e. runs the Web Administrator).

The default port number is 8031, but it may be any port number not in use on your JOB Server machine.

If the `Flags` option bitmask has the second bit set (see **"HTTP\_Server" on page 99**) the JOB Server starts listening on the specified port for HTTP requests in addition to acting in its normal role serving the JOB Client.

You may use the URL `http://machine_name:HTTPPort` where `machine_name` is the name (or IP address) of the JOB Server machine, and `HTTPPort` is the port number to communicate with the JOB Server from your browser.

- **Timeout**

The inactivity timeout in seconds (the default is 7200 - two hours).

The JOB Server starts a new thread (or process) for each client that connects and if there has been no communication in `Timeout` seconds the thread or process exits.

This ensures clients which fail to closedown properly do not cause increasing resource usage on the server.

To disable the timeout, set it to 0.



- **MaxThreadCount**

The maximum number of threads or processes the Easysoft JDBC-ODBC Bridge Server will allow at any time.

One thread/process is created for every ODBC connection. If `MaxThreadCount` is set to 0, there is no limit. The default is 0.

You can use this parameter to prevent too many simultaneous connections swamping your server.

- **RetryCount**

The number of times the JOB Server will attempt to create a thread or process to handle a connection, or the number of times the the server will attempt to obtain a license slot for a new connection.

- **RetryPause**

The time in seconds between each retry attempt (see ["RetryCount" on page 97](#)).

- **SingleStatement**

By default, the Easysoft JDBC-ODBC Bridge will attempt to pool statement handles in the ODBC driver and open new connections to support multiple concurrent statements, even if the backend ODBC driver does not.

If you know that you do not need this feature (i.e. you ensure that there is only a single result set open at any one time) set this to "Y".

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

- **HTTPAdmin**

The user name of the person allowed to make changes to the JOB Server via the Web Administrator.

This must be a valid user name in the operating system the server is running on. If set to the string "disabled" (omit the quotes) then authentication is not required (this does not mean the JOB Server stops authenticating incoming connections).

The value is case-sensitive.



**NB**

The "eye" symbol displayed against the HTTPAdmin parameter is used to denote that the value of the field is hidden from the user unless they gain access to the corresponding screen by clicking **Change** and entering user name and password details.

- **Path**

The installation path of the Easysoft JDBC-ODBC Bridge.

This is a read-only parameter for information only.

- **Logging**

A bitmask telling the JOB Server what sorts of event to record in the log file.

This should only be used as directed by Easysoft support and will slow the Easysoft JDBC-ODBC Bridge down considerably if set.

You may specify the number as decimal or hexadecimal (e.g.

2047 or 0x7ff).

- **LogDir**

The directory where log files are created (see `Logging`).

It defaults to `c:\temp` on Windows and `/tmp` on Unix.

- **Flags**

A bitmask of operational flags.

The bitmasks are split into check boxes, one for each bit in `Flags`:

- **Authentication\_Disabled** (0x1) If set authentication is disabled in the JOB Server. Setting this parameter should be considered as a security risk. However, on some high hit servers in a controlled environment where you do not need to authenticate the connections this can save a considerable amount of time during connections (e.g. in Windows NT, 0.25 to 0.75 seconds of the connection time can be spent in authentication). The default is off.
- **HTTP\_Server** (0x2) If set the JOB Server listens for HTTP connections to the Web Administrator. The default is on.
- **MultiProcess** (0x4) If set the JOB Server starts a new process rather than a new thread for each incoming connection. Use this setting for an ODBC driver which is not thread-safe or is leaking memory. The default is off.

<b>Unix</b>	The <b>MultiProcess</b> flag cannot be updated, as the server is not multi-threaded and therefore always starts new processes.
<b>Win 9x</b>	The <b>MultiProcess</b> flag is ignored.

- **HideSensitive** (0x10) If set the Web Administrator hides sensitive parameters on the **Configuration** page. The `HTTPAdmin` and `Authentication_Disabled` parameters are hidden when `HideSensitive` is set. All parameters are always shown in full on the change configurable parameters page as this page is password protected. The default is off.
- **ReverseLookup** (0x20) If set the JOB Server calls `gethostbyaddr()` on the connecting client's IP address to obtain the client's machine name. On machines where DNS is not set up properly this can cause problems and in any case adds time to the connection. `ReverseLookup` currently only affects the number of different clients page on the Statistics screen, where "unknown" will be displayed instead of the machine name for the machine names of connecting clients if `ReverseLookup` is off. The default is off.
- **AuditODBCAccess** (0x40) If set the JOB Server audits all connections to a log file which may be viewed through the Web Administrator. The audit file is written to the `LogDir` directory as `esjob_access.log`. The default is off.
- **ShowProcessTime** (0x100) If set user and kernel CPU times for the main JOB Server process are displayed on the Web Administrator **Statistics** screen (see [Server CPU time \(s\) on page 90](#)). Values are updated every five seconds. The default is off.

# CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

**Easysoft JDBC-ODBC Bridge Web Administrator**

**Configuration**

Port : 8831 ?

HTTPPort : 8031 ?

Timeout : 7200 ?

MaxThreadCount : 100 ?

RetryCount : 5 ?

RetryPause : 3 ?

SingleStatement : n ?

HTTPAdmin : John Smith ?

Path : C:\Program Files\Easysoft\Easysoft JDBC-ODBC Bridge

Logging : 64 ?

LogDir : C:\temp ?

Authentication\_Disabled :  ?

HTTP\_Server :  ?

MultiProcess :  ?

Reserved :  ?

HideSensitive :  ?

ReverseLookup :  ?

AuditODBCAccess :  ?

Reserved :  ?

ShowProcessTime :  ?

**Submit**

Data Access by Easysoft

Figure 37: The Web Administrator Change Configuration screen

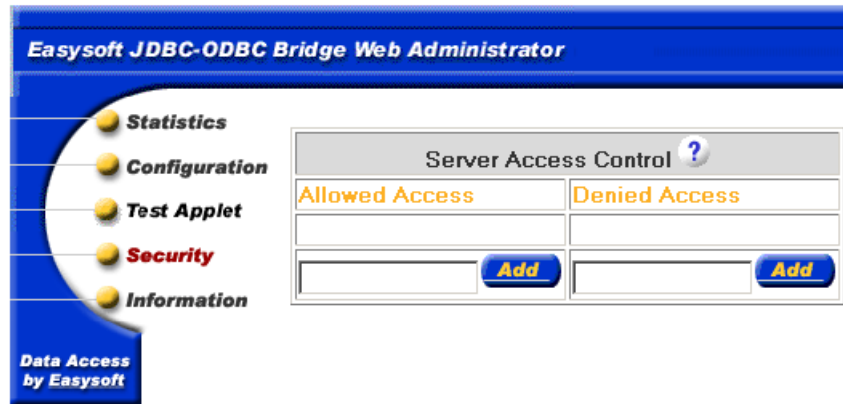
## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

### THE SECURITY SCREEN

The Web Administrator **Security** screen displays and allows the user to view or change the set of hosts that are allowed to connect to the Easysoft JDBC-ODBC Bridge.

The Web Administrator user name and password are required if they have not yet been entered:



**Figure 38: The Web Administrator Security screen**

In addition to the user names and passwords of your system and of the database management system, the Easysoft JDBC-ODBC Bridge provides another layer of security with access control lists.

To add an IP address to a list, type the address or address-stem into either the **Allowed Access** or **Denied Access** boxes and click **Add**.

Unix administrators will recognize this mechanism from the `hosts.allow` and `hosts.deny` files.

**NB**

Although the approach is similar, the rules for determining whether or not a host should be allowed to connect are different from those for `hosts.allow` and `hosts.deny`.

Security for the Easysoft JDBC-ODBC Bridge takes the form of two lists of IP addresses.

When a host attempts to connect to the JOB Server, access is only granted if:

- the `allowed` list is empty and the IP address is not in the `denied` list.
- the IP address is in the `allowed` list.

The lists can be edited either in the registry (Windows), their relevant flat files (Unix), or via the Web Administrator (Windows and Unix when run standalone.)

Addresses must be entered using the IP 'dot' notation.

Entries which consist of fewer than four fields represent all the addresses that match the fields which are defined.

e.g.

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

**NB**

Direct editing is not supported by Easysoft and will not allow access to changed values until the server has been rebooted.

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

### THE INFORMATION SCREEN

The Web Administrator **Information** screen displays a list of links to Easysoft support resources:

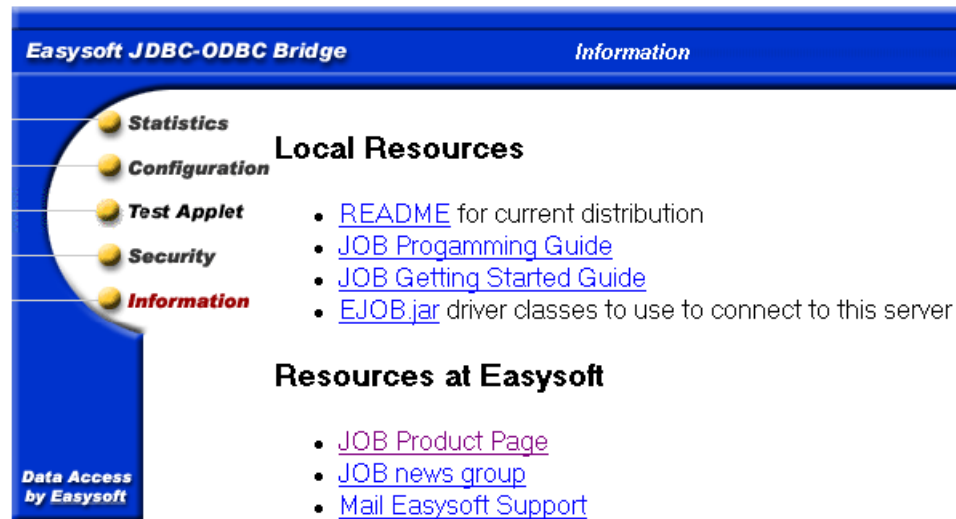


Figure 39: The Web Administrator Information screen

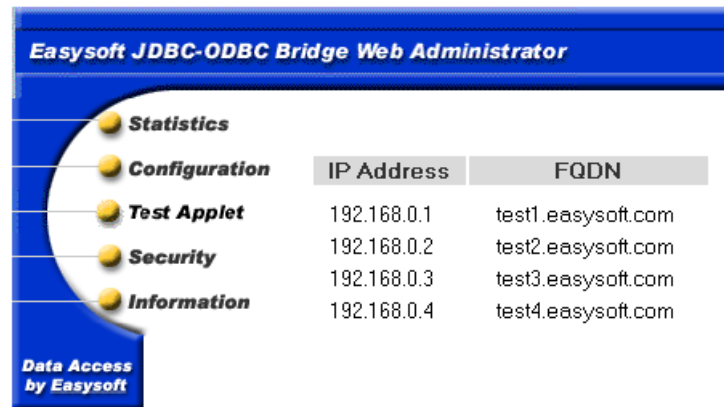
- **README** - A document listing any recent updates to the Easysoft JDBC-ODBC Bridge product.
- **JOB Programming Guide** - A guide for Java developers who wish to add JDBC connectivity to their applications using the Easysoft JDBC-ODBC Bridge.
- **JOB Getting Started Guide** - A guide to the download, installation and licensing of the Easysoft JDBC-ODBC Bridge, and to using the sample Java test program to access ODBC compliant data.



- **EJOB.jar** - Download a copy of the Easysoft JDBC-ODBC Bridge `EJOB.jar` file.
- **JOB Product Page** - The Easysoft JDBC-ODBC Bridge page on the Easysoft web site.
- **JOB news group** - A forum for posting and answering questions with other users concerning the Easysoft JDBC-ODBC Bridge.
- **Mail Easysoft Support** - A form allowing a user to email a support request for the Easysoft JDBC-ODBC Bridge.

## THE CLIENT HOSTS SCREEN

The Web Administrator **Client Hosts** screen displays a list of individual clients which have connected to the JOB Server. These connection attempts may not have been fully successful (e.g. not authenticated or denied access).



	IP Address	FQDN
Statistics	192.168.0.1	test1.easyssoft.com
Configuration	192.168.0.2	test2.easyssoft.com
Test Applet	192.168.0.3	test3.easyssoft.com
Security	192.168.0.4	test4.easyssoft.com
Information		

Figure 40: The Web Administrator **Client Hosts** screen

The screen contains the following fields:

- **IP Address** - The IP addresses of connecting clients.

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

- **FQDN** - The Fully Qualified Domain Name of connecting clients, which are displayed as "unknown", unless `ReverseLookup` is turned on (see "[ReverseLookup](#)" on page 100).

---

### The Windows Registry entry

If you cannot remember the user name or you do not know the corresponding password, then you will not be able to modify the configuration of the JOB Server via the Web Administrator.

In this case you need to log in directly to the server host and edit the registry.

The registry key for the Easysoft JDBC-ODBC-Bridge is:

HKEY\_LOCAL\_MACHINE/SOFTWARE/Easysoft JDBC-ODBC Bridge/Configuration/System/settings

Note that the registry key contains a string value for each of the Server Configurable Parameters (numeric parameters are stored in strings):

Name	Type	Data
(Default)	REG_SZ	(value not set)
Flags	REG_SZ	0x12
HTTPAdmin	REG_SZ	John Smith
HTTPPort	REG_SZ	8031
LogDir	REG_SZ	C:\temp
Logging	REG_SZ	0
MaxThreadCount	REG_SZ	10
Path	REG_SZ	C:\Program Files\Easysoft\Easysoft JDBC-ODBC Bridge
Port	REG_SZ	8831
RetryCount	REG_SZ	5

**Figure 41: The Windows Registry entry**

See "[The Configuration Screen](#)" on page 94 for full details of each of these settings.

---

## The Unix initialization file

If you do not know the password then you will not be able to modify the JOB Server configuration via the Web Administrator.

In this case, for the you need to log in directly to the server host and edit `<InstallDir>/easysoft/job/server/esjobserver.ini`.

If `<Installir>` is anything other than `/usr/local/` then there will be a symbolic link `/usr/local/easysoft` to the real Easysoft directory.

An example `esjobserver.ini` file follows:

```
{Settings}
Port = 8831
HTTPPort = 8031
Timeout = 3600
LogDir = /tmp
Logging = 0
Flags = 38
RetryCount = 5
RetryPause = 3
AllowList =
DenyList =
MaxThreadCount = 0
MaxClientConnect = 0
HTTPAdmin = John Smith
```

## CONFIGURING THE SERVER

*Configuring the Easysoft JDBC-ODBC Bridge Server*

See "[The Configuration Screen](#)" on page 94 for full details of each of these settings.

### **NB**

Lines beginning with # are comments. The first non-comment line should be {Settings} and the rest of the file should be *Key=Value* pairs. Bitmask values are given either in decimal or in hexadecimal. The `Flags` attribute must be included, and must be given a value of 38, unless you are advised otherwise by Easysoft.

# CONFIGURING THE CLIENT

---

## Configuring the Easysoft JDBC-ODBC Bridge Client

This section takes you through connecting to an ODBC data source from an existing JDBC program.

---

### Chapter Guide

- [Connecting from Java](#)
- [Tuning](#)

---

## Connecting from Java

In JDBC, you connect to data sources using an URL, the format of which is dependent on the driver being used.

The format for an Easysoft JDBC-ODBC Bridge URL is:

```
jdbc:easysoft://hostname:port/[RemoteDSN]{:attr=value}*
```

where `[ ]` denotes an optional item and `{ }*` denotes zero-or-more occurrences.

**NB**

You must omit an attribute altogether (rather than passing an empty string) if you do not wish it to apply.

*hostname* and *port* describe how to access the remote machine where the JOB Server is running and should be used in the same way as in the address box of a browser, except that the additional name `appletHost` is supported.

`appletHost` is understood by the JOB Driver class and refers to the server that the applet was loaded from. For an example of using `appletHost`, click the **Getting Started** link at <http://www.easysoft.com/products/job> and look at `OutputApplet.java`.

*RemoteDSN* identifies a System DSN (Data Source Name) configured on that machine.

The DSN should be set up with all the attributes it requires on the remote machine, with the possible exceptions of `UID` and `PWD` which can be specified on connection.

The following attributes are recognized:

`:user=Database User ID`

`:password=Database Password`

`:logonuser=Remote System User ID`  
`:logonpassword=Remote System Password`  
`:cleartext=on or off`  
`:limit=Row Limit`  
`:trace=on or off`  
`:unicode=on or off`

- `user` and `password` are given to the ODBC driver as `UID` and `PWD` and must be used if the back-end database requires authentication.
- `logonuser` and `logonpassword` specify an existing user on the target machine, but are only required if authentication is enabled on the JOB Server.

**NB** Set the `ClearText` attribute to true if you wish to be able to see the contents of these fields.

They may also be used if the database engine requires that the connecting process (i.e. the JOB Server) is logged on via a specific user.

## Caution!

Running the Windows service as a specified user (other than local admin) means that authentication must be disabled and that any user can then therefore connect, creating a security risk. You will also be unable to connect to trusted data sources (such as Microsoft SQL Server) as a system account.

- `cleartext` disables the default encrypted transmission of connection information (connection string, user names and passwords) over the network (useful for debugging purposes).

## CONFIGURING THE CLIENT

### *Configuring the Easysoft JDBC-ODBC Bridge Client*

- `limit` fixes the maximum number of rows that will be returned in any result set, limiting the time and space requirements that can be generated by any single query.

The limit is actually set in the ODBC driver by the JOB Server, so it is supported only if your ODBC driver supports it.

**NB**

The value of `Limit` cannot be changed during the lifespan of a connection, so it must be set either to a value which will accommodate the largest possible result set or be omitted (which means no limit is set).

- `trace` specifies whether the Driver (i.e. not the server) should write a detailed log to the Java Console.

If set to `off` or omitted then no trace file is written. In the trace, return values, procedure calls (beginning `*`) and Remote Procedure Calls (beginning `-->`) with their return values are logged.

**NB**

Note that setting the `trace` attribute slows down performance considerably.

- `unicode=on` specifies that all character data is retrieved as Unicode.

This setting must be specified in cases where back-end data sources support Unicode, but do not report that data conforms to the Unicode standard. The default is `unicode=on`, so that only columns containing Unicode data are retrieved as Unicode. Set to `unicode=off` to prevent any data from being retrieved as Unicode.



### **NB**

You do not have to set up a DSN in order to make a connection. Instead, you can specify one or more connection attributes by using the *attr=value* clause within the Easysoft JDBC-ODBC Bridge URL.

### **WHERE TO PUT EJOB.JAR**

Before you can connect your Java application to an ODBC data source, you must make the client JDBC driver (`EJOB.jar`) available to your application. You do this by doing either of the following:

- add `EJOB.jar` to your `CLASSPATH`
- unjar the `EJOB.jar` file into the directory where you are running your application, using the following command:

```
jar xvf EJOB.jar
```

This will create an `\easysoft` directory (and subdirectories) within your application directory.

### **CONNECTION DIFFICULTIES**

If you experience difficulties when connecting your Java applet or application to the Easysoft JDBC-ODBC Bridge, check the following:

- ensure you have added `EJOB.jar` to your `CLASSPATH` or have copied to the directory where you are running your application
- check that you are using the appropriate ports in the JDBC URL. The defaults are 8831 for the JOB Server and 8031 for its web server
- when connecting from an applet, check that you have specified `archive=EJOB.jar` in the `<applet>` tag.

For example:

```
<APPLET
ARCHIVE=EJOB.jar
CODE="OutputApplet.class"
WIDTH=250
HEIGHT=200>
<PARAM NAME = "jdbcUrl"
VALUE = "jdbc:easysoft://appletHost:8831/northwind">
</PARAM>
</APPLET>
```

- when connecting from an applet, if you receive a 'no suitable driver' error from the JDBC driver manager then the JDBC driver manager cannot find a JDBC driver that can recognise the JDBC URL with which you have tried to connect. This could be because the JDBC URL is not correct or because the driver classes are not loaded.

Try using the following syntax to resolve this:

```
Class.forName("easysoft.sql.jobDriver").newInstance();
```

- if you are running Windows 95/98, remember that you must start the JOB Server manually by selecting **Start > Program Files > Easysoft > Easysoft JDBC-ODBC Bridge > JOB Server** (see ["Starting the JOB Server on Windows" on page 77](#)).

For an example of connecting to the JOB from a Java applet, run a web browser and then:

- go to <http://demo.easyssoft.com/job/demo/OutputApplet.html> and view the source HTML file
- go to <http://demo.easyssoft.com/job/demo/OutputApplet.java> to view the sample code

Additional documentation and a connection example are provided with the Web Administrator, listening by default at port 8031.

To access this information, run a web browser and go to:

`http://serverhost:8031/`

where *serverhost* is the name of the host running the JOB Server. If the serverhost is your local machine, you can type:

`http://localhost:8031/`

**Getting Started** provides details on how to make a connection.

**Demonstration** invokes an applet which can be used to connect to ODBC data sources (see "[The Test Applet](#)" on page 115).

## THE TEST APPLET

Once you have an ODBC data source set up on the same machine as the JOB Server:

1. Run a browser and go to `http://serverhost:8031/demo`

where *serverhost* is replaced by the name of your host.

The demonstration applet runs automatically within the Web Administrator built in to the JOB Server.

## CONFIGURING THE CLIENT

*Configuring the Easysoft JDBC-ODBC Bridge Client*

2. In **Logon User** and **Logon PWD**, enter a valid user name and password for the server machine (these may be for either the server machine itself or for a domain, if the server is on a network):

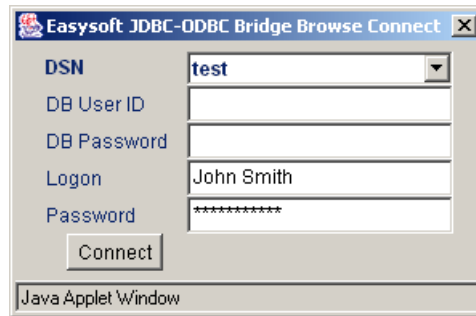
The screenshot shows a web-based logon interface. At the top, there are four input fields: 'URL:' with the value 'jdbc:easysoft:limit=100:unicode=on', 'Logon User:', 'Logon PWD:', and 'SQL:'. To the right of these fields are three buttons: 'Connect', 'Disconnect', and 'Submit'. Below the input fields are two empty panes labeled 'Tables' and 'Results'. At the bottom of the interface, there is a checkbox labeled 'Display Catalog and Schema' which is currently unchecked, and a text prompt 'Enter a URL and click on Connect...'.

**Figure 42: The Test Applet Logon screen**

3. Click **Connect**.

A dialog box is displayed listing the data source(s) available on the same machine as the JOB Server.

4. From the **DSN** drop-down list, choose the data source to which you want to connect:



**Figure 43: The Test Applet Connect screen**

5. In **DB User ID** and **DB Password**, enter the user name and password required by the back-end database (use `demo` and `easysoft` to connect to the `pubs` database at Easysoft).

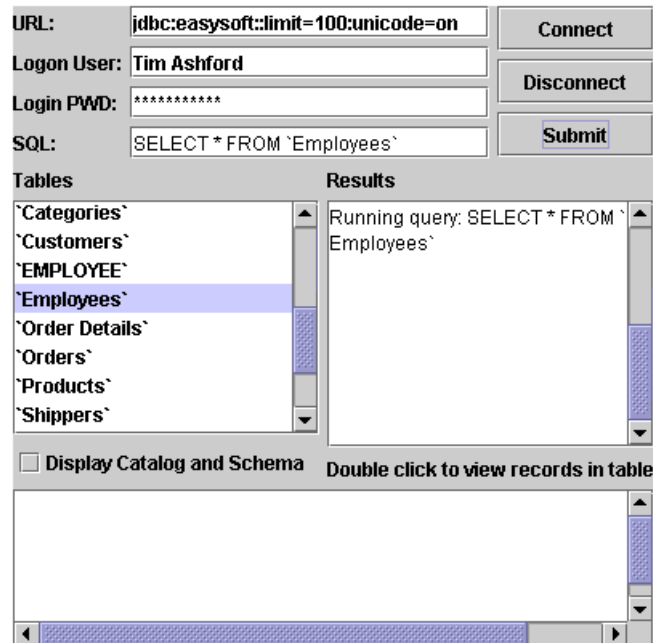
Note that the details entered in the **Logon User** and **Logon PWD** fields on the **Logon** screen are displayed in the **Logon** and **Password** fields on this screen.

6. Click **Connect** to make the connection.

## CONFIGURING THE CLIENT

*Configuring the Easysoft JDBC-ODBC Bridge Client*

If the connection is successful, the tables in the data source are listed in a new window.



**Figure 44: The Test Applet JobLet screen**

7. To execute a query on the database, either type a query into the **SQL** box and then click **Submit**.

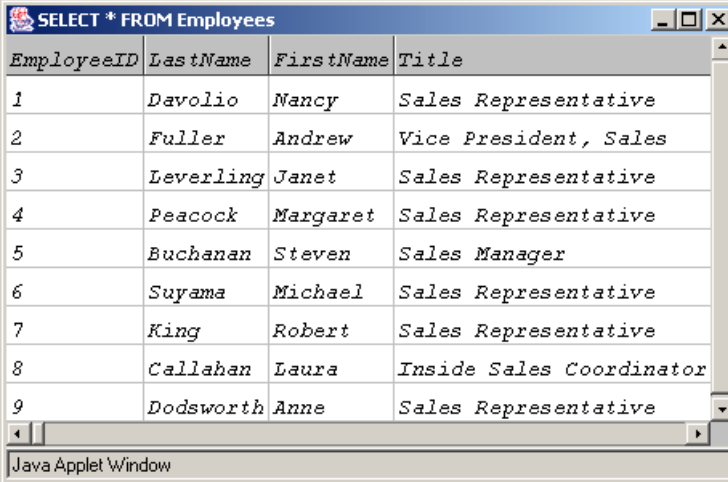
– OR –

Double-click on a table name in the list of **Tables** on the left-hand side of the window to retrieve up to 100 rows of that table.

**NB**

The number of tables and rows displayed is specified in the JDBC URL :`limit=100` connection attribute.

The query results are displayed in a new window:



The screenshot shows a Java Applet Window titled "SELECT \* FROM Employees". The window contains a table with the following data:

<i>EmployeeID</i>	<i>LastName</i>	<i>FirstName</i>	<i>Title</i>
1	<i>Davolio</i>	<i>Nancy</i>	<i>Sales Representative</i>
2	<i>Fuller</i>	<i>Andrew</i>	<i>Vice President, Sales</i>
3	<i>Leverling</i>	<i>Janet</i>	<i>Sales Representative</i>
4	<i>Peacock</i>	<i>Margaret</i>	<i>Sales Representative</i>
5	<i>Buchanan</i>	<i>Steven</i>	<i>Sales Manager</i>
6	<i>Suyama</i>	<i>Michael</i>	<i>Sales Representative</i>
7	<i>King</i>	<i>Robert</i>	<i>Sales Representative</i>
8	<i>Callahan</i>	<i>Laura</i>	<i>Inside Sales Coordinator</i>
9	<i>Dodsworth</i>	<i>Anne</i>	<i>Sales Representative</i>

The window also shows a status bar at the bottom that reads "Java Applet Window".

Figure 45: The Test Applet Results screen

---

## Tuning

### SIZE

The size of your Java applet affects download times, which can be critical in a consumer website.

The Easysoft JDBC-ODBC Bridge driver, in its compressed `.jar` file, is only 64k.

You could place the `.jar` file into the `CLASSPATH` on the client machines, so that it does not need to be downloaded at all, but this small benefit in reduced size is almost certainly outweighed by the loss of zero-installation functionality.

To reduce the size of your bytecode, try compiling the application or applet with Sun's `javac`, rather than whatever IDE you are using. Obfuscate the classes and place them in a `.jar` file.

### SPEED

If you are returning large queries, but are only interested in the first few records, try setting the *limit* attribute in the connection, which reduces network traffic by restricting the maximum number of rows that can be returned in any single result-set.

Note that it is important to set the attribute to a large enough value so that no important behavior will be lost and that this feature will normally only be of use while developing your program.

To further reduce network traffic, get the DBMS to perform as much of filtering as possible.

The network is probably more of a bottleneck than CPU or I/O resources on the server, so it is better to design relatively complex queries that return a minimal result set rather than simple, naive queries that are then processed on the client side.



# PROGRAMMING GUIDE

# 5

---

## Easysoft JDBC-ODBC Bridge Programming Guide

This section is aimed at Java developers who wish to add JDBC connectivity to their applications using the Easysoft JDBC-ODBC Bridge.

It is assumed that the JOB Server has been successfully installed and configured and that you know the **hostname** for the server to which you wish to connect.

---

### Chapter Guide

- **Introduction**
- **Java applications**
- **Java applets**
- **Making applets portable**
- **Loading JOB Driver classes**
- **Connecting to the JOB Driver**
- **Easysoft JDBC URL attributes**
- **Additional notes**
- **Easysoft JOB URL examples**
- **Minimal Java applet demonstration**

---

## Introduction

The Easysoft JDBC-ODBC Bridge comes in two components:

- The JOB Driver: a 100% Java JDBC 1.2/2.1 compliant driver.
- The JOB Server: a server built specifically for the platform where your target ODBC System Data Source is configured.

The JOB Server incorporates a Web Administrator (see **“The Web Administrator” on page 87**) that provides access to administration, monitoring and test facilities through a standard web interface.

The classes that make up the driver are in the Java archive file `EJOB.jar`. The driver class is called `easysoft.sql.jobDriver`.

---

## Java applications

Download `EJOB.jar` to the client machine from the JOB Server at `http://hostname:8031/jars/EJOB.jar` (where 8031 is the web server port).

On the client machine you must either amend or create the 'CLASSPATH' setting in the **Environment Variables**, which are found under **Start > Settings > Control Panel > System > Advanced** in Windows 2000 (this may vary with other versions of Windows).

If 'CLASSPATH' already contains a value, add a semi-colon and the full path and file name of your `EJOB.jar` file to the existing value.

To add a new 'CLASSPATH', click on the **Environment Variables** tab and click **New**.

Put 'CLASSPATH' in the Variable Name, the full path and file name in the Variable Value and then click **OK**.

Please note that you need to update either the User or System Environment Variable list as your application requires.

`http://hostname:8031/doc/output/TEST.java.html`  
contains sample code for a simple Java application.

The source may be saved from

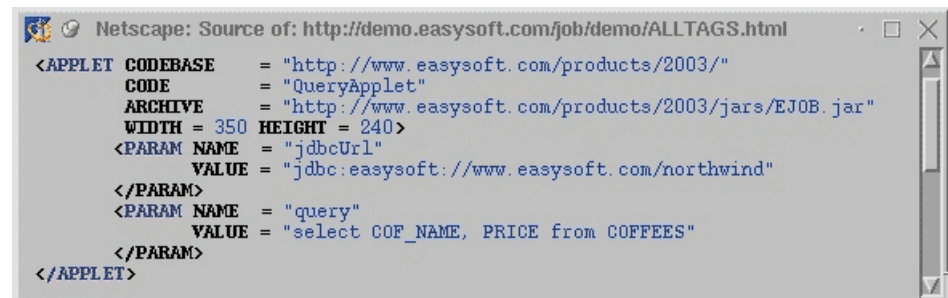
`http://hostname:8031/demo/TEST.java`, compiled and used to test the connection to an Easysoft JDBC-ODBC Bridge Server (assuming that 8031 is the Web Server port).

---

## Java applets

For applet deployment, place the `EJOB.jar` file with your applet in a directory readable by your web server and add the appropriate `ARCHIVE=EJOB.jar` attribute to the `APPLET` tag on your HTML page.

`EJOB.jar` can be downloaded from the link on the installed Easysoft JDBC-ODBC Bridge Web Administrator **Information** page:



```
<APPLET CODEBASE = "http://www.easysoft.com/products/2003/"
        CODE = "QueryApplet"
        ARCHIVE = "http://www.easysoft.com/products/2003/jars/EJOB.jar"
        WIDTH = 350 HEIGHT = 240>
<PARAM NAME = "jdbcUrl"
        VALUE = "jdbc:easysoft://www.easysoft.com/northwind"
</PARAM>
<PARAM NAME = "query"
        VALUE = "select COF_NAME, PRICE from COFFEES"
</PARAM>
</APPLET>
```

Figure 46: An example applet tag

---

## Making applets portable

In [Figure 46 on page 123](#) the `jdbcUrl` explicitly specifies the host on which the JOB Server is running.

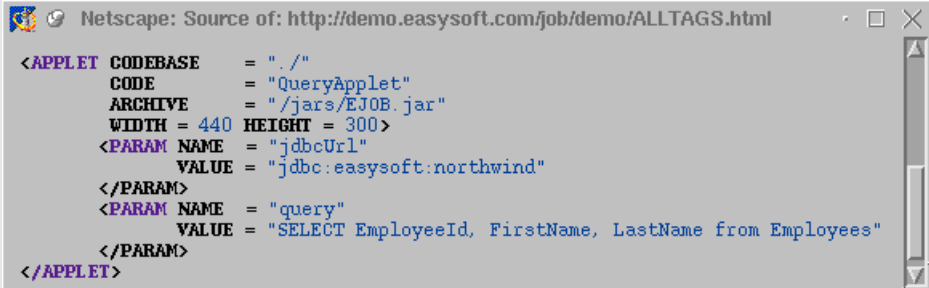
Applet restrictions dictate that applets can only open socket connections to machines from which they have been downloaded. There is no browser independent way of determining from where a `jar` file has been downloaded.

The Easysoft JDBC-ODBC Bridge includes a facility that enables a short form of the URL to be used, eliminating the need to specify the **hostname** in your applet tag or Java code.

To achieve this the Web server must be installed on the same machine as the JOB Server.

Ensure the file `ez.class` is placed in the same directory as the HTML file used to run your applet, or in the directory specified as the `CODEBASE` attribute in your applet tag.

The driver will now attempt to discover the name of the host that it is loaded from, and will attempt a connection to the default 8831 port.



```
<APPLET CODEBASE = "./"
        CODE = "QueryApplet"
        ARCHIVE = "/jars/EJOB.jar"
        WIDTH = 440 HEIGHT = 300>
<PARAM NAME = "jdbcUrl"
        VALUE = "jdbc:easysoft:northwind"
</PARAM>
<PARAM NAME = "query"
        VALUE = "SELECT EmployeeId, FirstName, LastName from Employees"
</PARAM>
</APPLET>
```

Figure 47: An example applet tag using `ez.class`

ez is a subclass of the `easysoft.sql.jobDriver` and can be used as an alternative short name in interactive applications which require you to enter the JDBC driver class name by typing `ez` instead of `easysoft.sql.jobDriver`.

You will find `ez.class` in `http://hostname/demo/ez.class`:

```
import easysoft.sql.*

public class ez extends jobDriver {

    public ez() throws SQLException {

    }

}
```

---

## Loading JOB Driver classes

You can load the JOB Driver classes in your code using:

```
Class.forName("easysoft.sql.jobDriver").newInstance();
```

Alternatively you could use:

```
Class.forName("ez").newInstance();
```

which in turn will instantiate the `easysoft.sql.jobDriver` class.

---

## Connecting to the JOB Driver

An example connection to an URL with the Easysoft subprotocol:

```
String jobUrl= "jdbc:easysoft://demo.easysoft.com/pubs";
```

```
Connection con = DriverManager.getConnection(jobUrl);
```

The general form of a JOB URL is:

```
<url> ::= jdbc:easysoft:[<server spec>][<database>]
        {:<attribute>=<value>}*
<server spec> ::= //[<host name>][:<port>]/
<database> ::= <dsn> | DSN=<dsn> | FILEDSN=<filedsn>
               <DSNlessconnection string >
```

where | separates optional items, [ ] denotes an optional item and { }\* denotes zero or more occurrences.

### **DSNLESS CONNECTIONS**

```
jdbc:easysoft://<host>/Driver={<driver
name>}[;<attr>=<value>]+
```

e.g.

```
jdbc:easysoft://<host>/Driver={Microsoft Access Driver
(*.mdb)};DBQ=C:\tsmall.mdb;trace=on
```

Note that when using DSNless connections the JOB JDBC URL attribute separator ':' is changed to ';' so that ':' can be used as part of the ODBC connection attributes (as in Windows driver names).

---

## Easysoft JDBC URL attributes

Attribute	Description
:user	Database User ID
:password	Database Password
:logonuser	Remote System User ID
:logonpassword	Remote System Password
:clearText=on off	Disable the default encrypted transmission of connection information (connection string, user names and passwords) over the network (useful for debugging purposes)
:limit = n	The maximum number of rows returned in any result set
:trace=on   off	Turn on/off tracing to stdout in the client
:traceFile=<filespec>	Turn on tracing and redirect output to a named file. Note that `:` is used as the attribute separator in JOB URLs. Hence you need to use the escape characters %3A for `:` in the file specs for <code>traceFile</code> attributes on Windows, e.g.  <pre>:traceFile=C%3A\logs\job.log</pre>
:multi=on	Prefix each line in a trace with the thread ID
:fetchSize=0	Turn off retrieval of columns in blocks
:unicode=on   off	Turn on retrieval of Unicode data from Unicode fields

**Figure 48: Easysoft JDBC URL attributes**

---

### Additional notes

**host name** and **port** describe how to access the remote machine where the JOB Server is running.

They must be explicitly specified in the JDBC URL if you wish to connect to a remote JOB Server.

**user** and **password** are given to the ODBC driver as UID and PWD. They must be used if the database engine requires authentication.

**logonuser** and **logonpassword** specify an existing system user on the target machine.

They must be used if the database engine requires the connecting process (in this case the JOB Server) to be logged in as a particular user..

### Caution!

Running the Windows service as a specified user (other than local admin) means that authentication must be disabled and that any user can then therefore connect, creating a security risk. You will also be unable to connect to trusted data sources (such as Microsoft SQL Server) as a system account.

**limit** is used to restrict the number of rows that will be returned in any result set.

The limit is actually set in the ODBC driver by the JOB Server as statement attribute `SQL_ATTR_MAX_ROWS`, so it is supported only if your ODBC driver supports it.

**fetchSize** should be set to zero to avoid the retrieval of far more columns (the default is 4k) in a results set than will actually be needed by the application.



---

## Easysoft JOB URL examples

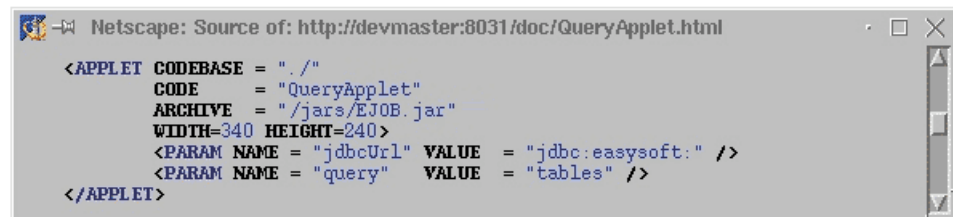
JOB Server	Platform
<code>jdbc:easysoft:northwind</code>	Driver will connect to the host from which the <code>ez.class</code> resource is obtained. This could be the local host for applications or the host from which the applet is downloaded, and will establish a connection to named ODBC data source 'northwind' in the same way as the Sun JDBC bridge.
<code>jdbc:easysoft:DSN=northwind</code>	Equivalent to <code>jdbc:easysoft:northwind</code> .
<code>jdbc:easysoft:</code>	Driver will connect to the host as with <code>jdbc:easysoft:northwind</code> . Since no database is specified, a list of available DSNs will be displayed in a dialog box.
<code>jdbc:easysoft:FILEDSN=myfiledsn</code>	As with <code>jdbc:easysoft:northwind</code> , but connecting to a file DSN data source, rather than a system DSN data source.
<code>jdbc:easysoft://:8899/northwind</code>	Connect the JOB Server to a non-default port.
<code>jdbc:easysoft://demo.easysoft.com/pubs</code>	In an application connect to the JOB Server at Easysoft.

**Figure 49: Easysoft JOB URL examples**

---

## Minimal Java applet demonstration

This is a very simple demonstration of an applet connecting to an ODBC data source using the Easysoft JDBC-ODBC Bridge. The applet tag used is shown here:



```
<APPLET CODEBASE = "./"
        CODE = "QueryApplet"
        ARCHIVE = "/jars/EJOB.jar"
        WIDTH=340 HEIGHT=240>
  <PARAM NAME = "jdbcUrl" VALUE = "jdbc:easysoft:" />
  <PARAM NAME = "query" VALUE = "tables" />
</APPLET>
```

Figure 50: A minimal Java applet demonstration

The `CODE` tag specifies the `QueryApplet.class` class file for the applet's code.

The `ARCHIVE` tag specifies the `"/jars/EJOB.jar"` Java archive file that contains the classes for the Easysoft JDBC-ODBC Bridge.

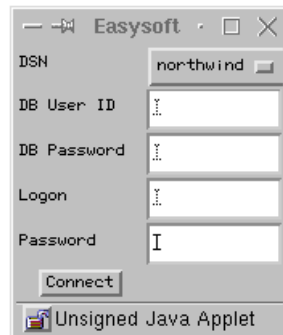
The applet takes two parameters:

- The JDBC URL required to connect to an ODBC data source on the machine where the JOB Server is installed and from which this HTML page is served.
- The JOB SQL pseudo-query to be executed on the data source.

You can view the source in HTML format at

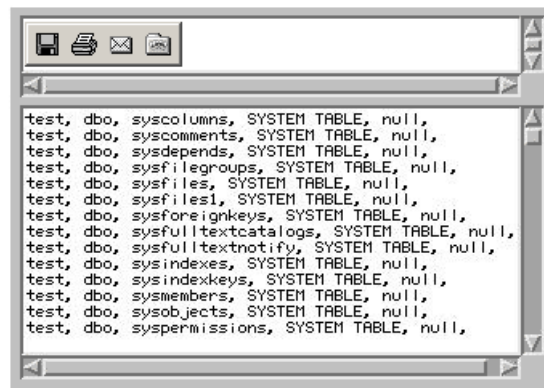
`http://localhost:8031/doc/output/QueryApplet.java.html` OR  
read and download (by right clicking and selecting **Save Target As**)  
the `http://localhost:8031/demo/QueryApplet.java`. These URLs  
assume that 8031 is the web server port.

When the page is loaded the applet is started and if the JOB Server is running it will popup a dialog box allowing a selection of available ODBC data sources to which to connect:



**Figure 51: The ODBC data source selection dialog box**

Upon successful connection, the result of running the pseudo-query **tables** is displayed:



**Figure 52: Displaying the result of the tables pseudo-query**

You can run the applet at

`http://localhost:8031/doc/QueryApplet.html`.

You may like to edit the HTML that launches the applet to specify your own query and JDBC URL parameter.

Use **File > Save As** from your browser on the page that runs the applet.

# TECHNICAL REFERENCE



---

## Technical Reference for the Easysoft JDBC-ODBC Bridge

This section contains extra information relating to the deployment of the Easysoft JDBC-ODBC Bridge.

---

### Appendix Guide

- **Java runtimes and browsers**
- **Windows platforms**
- **Unix platforms**
- **JDBC versions supported**
- **ODBC version required**
- **Supported data types**
- **Classpath**
- **Tracing**
- **Unicode**
- **Firewall and security issues**
- **Threading**

## TECHNICAL REFERENCE

*Technical Reference for the Easysoft JDBC-ODBC Bridge*

---

### Java runtimes and browsers

The Easysoft JDBC-ODBC Bridge Driver is built with JDK 1.3.

Applications require the Java Runtime Environment (JRE) 1.2 or higher.

Applets require Netscape Navigator 5.x (or higher) or Microsoft Internet Explorer 5.x (or higher).

---

### Windows platforms

The Easysoft JDBC-ODBC Bridge runs as a service under Windows NT 4, Windows 2000 and Windows XP.

The Easysoft JDBC-ODBC Bridge will also run under Windows 9x (where services are not supported) when started manually as an application.

#### Win 95

To run the Easysoft JDBC-ODBC Bridge on Windows 95, you need WinSock2, which can be downloaded from [http://www.microsoft.com/windows95/downloads/contents/wu/admintools/s\\_wunetworkingtools/w95sockets2/](http://www.microsoft.com/windows95/downloads/contents/wu/admintools/s_wunetworkingtools/w95sockets2/).

---

### Unix platforms

Log onto the Easysoft JDBC-ODBC Bridge Platforms page on the [Easysoft](#) website for details of currently available releases.

---

### JDBC versions supported

In addition to support for all JDBC 1.22 methods, the Easysoft JDBC-ODBC Bridge also supports all JDBC 2.1 functions (such as

batch updates, scrollable results sets, programmatic inserts, deletes, and updates), except SQL 3 data types.

---

### **ODBC version required**

The JOB Server makes ODBC 2.0 and ODBC 3.0 calls.

The use of a driver manager means that you can connect it to ODBC 2.0 drivers and the driver manager will remap the calls.

As Easysoft recommend the Microsoft driver manager for Windows and the unixODBC driver manager for Unix, if you decide to use a different driver manager then you need to be sure that it supports ODBC 3.0 and ODBC 2.0 calls.

---

### **Supported data types**

All standard JDBC data types are supported.

---

### **Classpath**

The default Easysoft JDBC-ODBC Bridge driver classes are found in:

`C:\Program Files\Easysoft\Easysoft JDBC-ODBC Bridge\EJOB.jar`  
(Windows)

or

`/usr/local/easysoft/job/EJOB.jar`

(Unix)

<b>NB</b> Alternate directories may be specified during installation.
---

This location must be added to the classpath in order to compile under Sun's JDK or run under the JRE.

For further information about driver classes and class paths, see <http://java.sun.com/j2se/1.4/docs/tooldocs/windows/classpath.html> (Windows) or <http://java.sun.com/j2se/1.4/docs/tooldocs/solaris/classpath.html> (Unix).

---

## Tracing

### CLIENT

The standard Driver Manager methods:

```
DriverManager.setLogStream(java.io.PrintStream(java.io.PrintStream out))
```

(for JDBC 1 driver managers)

or

```
DriverManager(java.io.PrintWriter out)
```

(for JDBC 2 driver managers)

can be used in your code to enable or disable tracing to a specified output file.

It is also possible to turn on tracing to the standard output (Java console) by adding the `:trace=on` attribute to the Easysoft JDBC URL.

Set the `clearText=on` attribute to disable the default encrypted transmission of connection information, which can make the tracing output unreadable.



For multithreaded applications and applets you should also set the `:multi=on` attribute, so that each line of the generated trace is prefixed with an indication of the thread that produced that line.

e.g.

```
jdbc:easysoft://host/dsn:trace=on:multi=on
```

## **S E R V E R**

Use ODBC tracing on the machine where the JOB Server is installed.

On Windows :

Start the ODBC driver manager administration interface via **Start Menu>Control Panel>ODBC Data Sources**, click on **Tracing**, ensure the specified filename is valid and click **Start Tracing Now**.

See [ODBC Tracing](#) for further details.

On Unix:

If you are using the unixODBC driver manager then tracing is enabled in the `odbcinst.ini` file (usually `/etc/odbcinst.ini`).

To enable tracing you must add two attributes to the [ODBC] section (if you do not have an [ODBC] section, create one):

```
Trace = Yes
TraceFile = /path/filename
```

e.g.

```
[ODBC]
Trace = Yes
TraceFile = /tmp/sql.log
```

Tracing will only work if the user who is running the application to be traced has write permission to `TraceFile` (and to the directory containing it).

---

## Unicode

Support for Unicode data in the Easysoft JDBC-ODBC Bridge can be enabled by setting the JDBC URL connection attribute `:unicode=on` (see [“Connecting from Java” on page 110](#)).

Use one of the two `getString` methods in the `ResultSet` class to read Unicode data and the `setString` method in the `PreparedStatement` class to write Unicode data.

Read and write statements to metadata table, row and column names containing Unicode characters are not supported, unless those names are restricted to ASCII characters only.

The `executeQuery` method in the `Statement` class does NOT support SQL containing Unicode character strings.

**NB**

Unicode data can also be read by using one of the two `getUnicodeStream` methods in the `ResultSet` class and written by using the `setUnicodeStream` method in the `PreparedStatement` class in addition to the `getString` and `setString` methods.

---

## Firewall and security issues

JOB Server connections are made through the port designated in the connection URL and any firewall must allow TCP protocol traffic on this port.

If the Java client program is an applet then the browser of a user will normally restrict network connections to the host from which the applet was served.

Note that the JOB Driver recognizes a special hostname in the URL, `appletHost`, which always refers to the host that the applet came from.

Using this hostname means that the name of the server may be changed or the ODBC data source and applet moved without breaking the reference.

For an example of using `appletHost`, follow the instructions in the **Getting Started** link at <http://www.easysoft.com/products/job> and look at `OutputApplet.java`.

For general use in browsers, the applet must be stored on the machine where the ODBC data source is set up. If this is not the web server then for low-traffic applications the Web Administrator built into the JOB Server may be used to serve the single page that contains the `<APPLET>` tag.

For heavier-duty applications it may be advisable to install a fully-fledged web server (such as Apache) on the machine holding the data sources.

Although the JOB Server and ODBC Driver must both be on the machine that serves the applet, the actual database does not have to be running on that same machine. Oracle, DB2, Informix and SQL Server are all examples of such network-enabled databases.

This applet restriction can normally be overcome by providing a signed applet, which a user may choose to "trust", thus allowing access to a wider range of network hosts.

---

## Threading

### CLIENT

The JOB Client is safe to use in multi-threaded applications and applets. It fully meets the requirements of the ODBC 3.5 specification, with the proviso that the ODBC data source must support asynchronous statement execution (see <http://java.sun.com/products/jdbc/driverdevs.html>).

All operations on `java.sql` and `javax.sql` objects are required to be multi-thread safe able to cope correctly with having several threads simultaneously calling the same object.

In other words, a statement execution in one thread should not block an execution in another thread. In particular, JDBC drivers should operate correctly when used from multiple threads.

An example of a specific use of multi-threading is the way a long-running statement can be cancelled. This is done by using one thread to execute the statement and a second one to cancel it with the method `Statement.cancel`.

Although in practice most JDBC objects are accessed in a single-threaded way, there needs to be support for multi-threading.

Some database APIs, such as ODBC, provide mechanisms for allowing SQL statements to execute asynchronously. This allows an application to start up a database operation in the background and then handle other work (such as managing a user interface) while waiting for the operation to complete.

As Java is a multi-threaded environment, there is no need to provide support for asynchronous statement execution. Java programmers can easily create a separate thread if they wish to execute statements asynchronously with respect to their main thread.

Some drivers may allow more concurrent execution than others, but developers should be able to assume fully concurrent execution. If the driver requires some form of synchronization, then the driver should provide it. In this situation, the only difference visible to the developer should be that applications run with reduced concurrency.

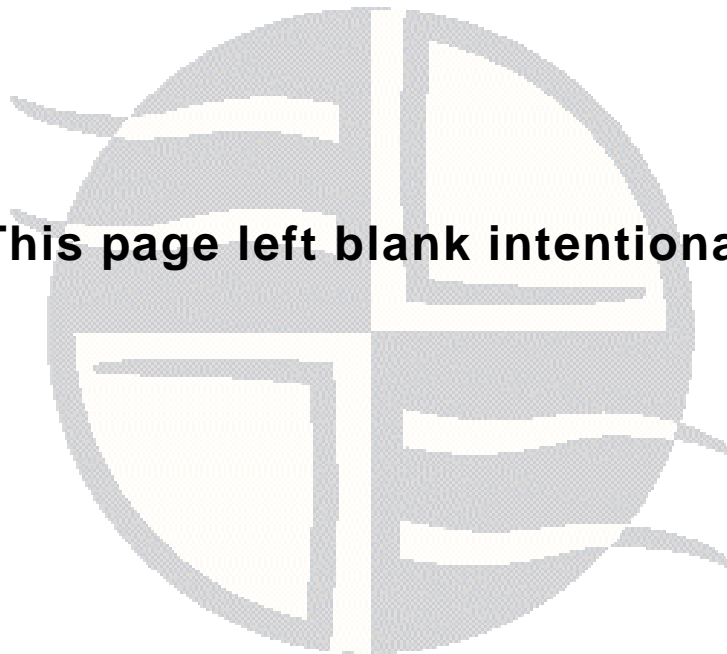
For example, two Statement objects on the same connection can be executed concurrently, and their result sets can be processed concurrently (from the perspective of the developer). Some drivers will provide this full concurrency, while others may execute one statement and wait until it completes before sending another one.

### SERVER

Under Windows, JOB Server may run in either multi-threaded or multi-process mode:

- In the default multi-threaded mode, the JOB Server creates a thread for each connection from a JOB Client.
- In multi-process mode, the JOB Server creates a process for each connection from a JOB Client (see "[MultiProcess](#)" on [page 99](#)).
- If the ODBC driver you are using is not thread-safe (e.g. Microsoft Access Jet versions prior to 4.0, as documented in the [Easysoft JDBC-ODBC Bridge FAQ](#)), run the JOB Server in multi-process mode.
- Under non-Windows platforms, the JOB Server is not required to be thread-safe, because a new process is started to handle each connection, either by the JOB Server in standalone mode (without `inetd`) or by `inetd` itself (if JOB is started as an `inet` service). There can only be one thread of execution in these circumstances.

**This page left blank intentionally**



# JDBC CERTIFICATION

A large orange square in the top right corner of the page, containing a white, serif capital letter 'B'.

---

Keylabs Certification Acceptance Report

---

The KeyLabs logo, featuring the word 'KeyLabs' in a bold, sans-serif font. 'Key' is in blue and 'Labs' is in black, with a small trademark symbol (TM) to the right.

## JDBC Certified

### Certification Acceptance Report

---

**Date:** 6th-Jan-2003

Congratulations! The product you have submitted to KeyLabs has successfully met the compliance criteria and has been accepted. Your product is now officially "JDBC Certified"! Below, you will find specific details concerning all of the testing performed.

#### VENDOR AND PRODUCT DESCRIPTION

**Vendor:** Easysoft Limited

**Product:** Easysoft JDBC-ODBC Bridge

**Description:** The Easysoft JDBC-ODBC Bridge allows JDBC access to any remote ODBC datasource.

### **PLATFORMS TESTED**

The product submitted was successfully tested for certification compliance on each of the following platform(s) and environment(s):

- Sun Ultra 10 using Solaris 8, CDE 1.4, Solaris 8 with Recommended Cluster Patches

### **CERTIFICATION TOOLS USED FOR TESTING**

- Java Platform Tested Against: JDK 1.4
- J2EE Version Used: 1.3.1
- JavaTest Version Used: 2.1.5

### **JAVATEST TESTCASE RESULTS**

- All JavaTest JDBC Testcases completed successfully.

Copyright 2003, KeyLabs, All rights reserved. The KeyLabs logo and logotype are trademarks of Lab Acquisition Corp. All other brand or product names are trademarks or registered trademarks of their respective holders.

### **Keylabs**

385 South 520 West

Lindon, UT 84042

Phone: 801-226-8200

Fax: 801-226-8205

E-mail: [jdbc-certified@keylabs.com](mailto:jdbc-certified@keylabs.com)



# GLOSSARY



---

## Terms and definitions

### **API**

Application Programmer Interface. An API is a published set of function calls and constants allowing different programmers to utilize a ready-written library of subroutines.

### **Applet**

A Java program that has certain restrictions placed upon it, such as restricted network and filesystem access. Applets are able to run in a web browser.

### **Application**

By contrast to an Applet, a Java Application has unrestricted network and file access.

### **Authorization code**

You must have an authorization code for the Easysoft product you wish to license in order to obtain a purchased license. When you purchase a product your authorization code is emailed to you. You do not need an authorization code to obtain a trial license.

### **Bitmask**

A value which, when written out in binary, has a meaning assigned to each digit, which can be 0 or 1. This is a very efficient way of storing a number of *flags* in a small amount of memory. When viewed in decimal it is a single number resulting from adding up the values of the individual bits, 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 (the *client*) is responsible for formatting and presenting the data. The client requests queries or actions be performed on the data by the server. Often these processes run on different *hosts* across a local-area network.

**Column**

The vertical dimension of a table. Columns are named and have a *domain* (or *type*). The term *column* might refer to only the *definition* of a column (i.e. its name and type), or to all the data in it.

**Connection String**

ODBC *driver managers* accept a connection string when a client connects. Ideally it contains all necessary attribute values to make the connection to a data source, but provision is made for the driver to negotiate with the application or the user for any missing information.

**Data source**

In the context of JDBC or ODBC, a data source 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 Object" on page 149](#)**.

**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**

In the context of JDBC or ODBC, software that accesses a proprietary data source, providing a standardized view of the data to ODBC.

**Driver Manager**

Software whose main function is to load drivers. ODBC or JDBC applications request a DSN through the Driver Manager. 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. A name associated with an ODBC data source. Driver Managers, such as unixODBC or the Microsoft Windows Driver Manager, use the Data Source Name to cross-reference configuration information and load the required driver.

**Field**

A placeholder for a single datum in a record. For example, you can have a Surname field in a Contact Details record. Called a *cell* in Microsoft Access.

**Flags**

Single-bit values, representing 'Yes' or 'No'. Also called "True" and "False".

**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 Web Administrator.

**JDBC (Java DataBase Connectivity)**

A Java *API* for database access based on the *ODBC* API, but includes an object-oriented interface to the underlying data source.

**JDK (Java Development Kit)**

A software development environment for writing Java applets and applications.

**JRE (Java RunTime Environment)**

A subset of the Java Development Kit (JDK) which supports the execution (but not the development) of Java applications and consists of the Java Virtual Machine (JVM), the core classes, and supporting files.

**JVM (Java Virtual Machine)**

Software that interprets and executes the byte codes in Java class files like a microprocessor would execute machine code. There are many virtual machines available from different vendors and for different purposes.

**Middleware**

Software that is placed between the *client* and the *server* to improve or expand functionality.

## **ODBC (Open DataBase Connectivity)**

A standard *API* for connecting application programs to relational database systems through a suitable *driver*. ODBC is available on a wide number of platforms and the Easysoft JDBC-ODBC Bridge allows the database and the application to reside on different machines across the network.

## **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 system running Solaris are three different platforms.

## **Server**

A computer, or *host*, on the network, designed for power and robustness rather than user-friendliness and convenience. Servers typically run round-the-clock and carry central corporate data.

– OR –

A process performing the centralized component of some task, for example extracting information from a corporate database. See also **“Client/Server” on page 146.**

## **Shared Object**

A piece of object code (i.e. a program fragment) that can be loaded and executed by other programs.

**SQL (Structured Query Language)**

A standard language for interacting with relational database systems, based on Relational Theory.

**System data source**

In the context of ODBC under Microsoft Windows, a data source which can be accessed by any user on a given system. See also **“User data source” on page 150.**

**Table**

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

software	
vendor	name
Easysoft	Easysoft JDBC-ODBC Bridge
Mysoft	My JDBC Application

This table has two columns; `vendor`, and `name`. It has two rows: one corresponding to Easysoft JDBC-ODBC Bridge and the other corresponding to MySoft’s Java application. The term *table* can also apply to just the definition of the table, without its data.

**User data source**

An ODBC Data Source with access limited to a specific user on a given system. See also **“System data source” on page 150.**

**Web Administrator**

A web server run by the Easysoft JDBC-ODBC Bridge which can be used to display, amend and configure selected parameters.

# INDEX

## A

---

access control .....	102
allowed list .....	103
Apache .....	139
API .....	145
Applet .....	145
applehost .....	139
Application .....	145
attributes	
for OOB Client DSNs on Unix .....	81-??
Authorization code .....	145

## B

---

beta releases .....	34
bitmask .....	145
in esjobserver.ini .....	108
bunzip .....	56

## C

---

Caution box .....	13
CD .....	34
CLASSPATH .....	35, 113, 120, 122, 135-136
client-server .....	146
connecting .....	110
connection difficulties .....	113
connection example .....	115
Control Panel .....	78
ODBC .....	72

## D

---

data source .....	146
-------------------	-----

# INDEX

*Easysoft JDBC-ODBC Bridge*

creating (Unix) .....	81
creating (windows) .....	75
data source administrator	
see driver manager	
demonstration applet .....	52, 65, 89, 130
denied list .....	103
documentation .....	34
driver manager .....	61, 147
DSN .....	147
DSNless connections .....	126
DSNs	
connecting from Java .....	110-112
creating on Unix .....	82-85
creating on Windows .....	72-77
defining .....	62
demonstration applet .....	117
testing on Unix .....	85-86

## E

---

EJOB.jar .....	35, 105, 113, 122-123, 130, 135
esjobserver.ini .....	107

## F

---

files	
/etc/odbc.ini .....	81
odbcinst.ini .....	82, 137
FTP .....	34

## G

---

gunzip .....	56
gzip .....	37

## H

---

history of ODBC .....	18
-----------------------	----



hosts.allow .....	103
hosts.deny .....	103

## **I**

---

installation	
requirements .....	55
requirements for Unix .....	55
installing	
on Unix .....	55
on Windows .....	38
isql .....	85

## **J**

---

JDBC driver types .....	20-23
-------------------------	-------

## **L**

---

license	
authorization code .....	47

## **N**

---

northwind database .....	72
Note box .....	13
NT services .....	78

## **O**

---

ODBC Data Source Administrator .....	73
ODBCConfig .....	80, 82
Open DataBase Connectivity .....	149

## **P**

---

patches .....	34
Platform note .....	13
proxy server .....	52, 65

# INDEX

*Easysoft JDBC-ODBC Bridge*

## R

---

Reference box ..... 13

## S

---

security ..... 102, 138

server setup

    Unix ..... 79

    Windows ..... 72

services

    NT ..... 78

Structured Query Language (SQL) ..... 150

Sun ..... 20-23

symbolic link ..... 107

## T

---

tar ..... 56

test applet ..... 115-119

threading ..... 140

Tracing ..... 136

## U

---

uncompress ..... 56

Unicode ..... 111-112, 127, 138

uninstalling

    on Unix ..... 70

    on Windows ..... 53

unixODBC ..... 28

    distribution ..... 27, 35

    installation ..... 61-62

    isql ..... 85

    ODBC Data Source Administrator ..... 82

    ODBC version support ..... 135

    using with JOB ..... 80-82

upgrades ..... 34

**W**

---

Web Administrator .....	28, 35
Change screen .....	95
Client Hosts screen .....	105
Configuration screen .....	94
Information screen .....	104
Security screen .....	102
Statistics screen .....	90
web site .....	34
Windows 95 .....	37, 134
WinSock2 .....	37, 134

**X**

---

X server .....	80
----------------	----