

Easysoft JDBC- Access Gateway User's Guide

This manual documents version 1.1.n of the Easysoft JDBC-Access Gateway.

Copyright © 1993-2025 Easysoft Limited.

All rights reserved.

You may not reverse engineer, decompile, or disassemble this manual. Information in this document 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.

Table of contents

Getting started	4
Overview	5
Installing the Easysoft JDBC-Access Gateway	6
Installing on Windows	6
Uninstalling on Windows	8
Connecting to your Access database	9
Registering the Easysoft JDBC-Access Gateway	9
The connection URL	9
Tracing	11
Java Runtime Environment (JRE) requirements	12
Setting the class path	13
Data types	14
Cursors	16
TYPE_FORWARD_ONLY Example	16
TYPE_SCROLL_INSENSITIVE Example	17
Unicode	18
Background	18
Unicode and the Easysoft JDBC-Access Gateway	18
Index	20

Getting started

- [Overview](#)
- [Installing the Easysoft JDBC-Access Gateway](#)
- [Connecting to your Access database](#)
- [Tracing](#)

Overview

The Easysoft JDBC-Access Gateway provides JDBC access from Java applications, application servers, and servlets to MDB or ACCDB files. If you configure your Java security policy to allow the Easysoft JDBC-Access Gateway to load its native component, the driver can also be used with Java applets.

Access has four components: a structure to hold data (tables), a way to manipulate that data, an environment to create a front end for the data (Design view of forms and reports) and tools that can run the front end (Data view of forms and reports). Access data is stored in a database file, which has the extension .mdb or .accdb.

The Easysoft JDBC-Access Gateway is a JDBC driver for Access that uses the Java Native Interface (JNI) to communicate with Microsoft's Access ODBC driver library. The Access ODBC driver is used to connect to the target database file.

The Access ODBC driver must be installed on the machine on which you install the Easysoft JDBC-Access Gateway. Since Windows 2000, the Access ODBC driver (one of the Jet Database Engine Components) has shipped with Windows and so should be already present on your machine as part of the Windows installation.

If the file you want to connect to is an ACCDB format database (.accdb) you need to install aceodbc.dll on the Easysoft JDBC-Access Gateway machine, which is available from Microsoft's web site.

The target database file must be visible through the local file system on the Easysoft JDBC-Access Gateway machine. For example, in a folder on this machine or a mapped network drive or in a shared folder.

Installing the Easysoft JDBC-Access Gateway

Installing on Windows

The Windows installation can be done by anyone with local administrator privileges.

1. [Download the Easysoft JDBC-Access Gateway installer.](#)
2. Follow the onscreen instructions to progress through the installation wizard.

Updating files that are in use

To avoid rebooting your computer, the Easysoft JDBC-Access Gateway installer prompts you when files that it needs to update are in use by another application or service. This frees the locked files and allows the installation to complete without a system restart. The installer uses the **Restart Manager** to locate the applications that are using files that need updating. These applications are displayed in the **Files in Use** dialog box. To avoid a system restart, choose **Automatically close applications and attempt to restart them after setup is complete**. The Easysoft JDBC-Access Gateway installer then uses **Restart Manager** to try to stop and restart each application or service in the list. If possible, **Restart Manager** restores applications to the same state that they were in before it shut them down.

Licensing

By default, the installer starts the Easysoft License Manager, because you can't use the Easysoft JDBC-Access Gateway until you have a license. If you choose not to run Easysoft License Manager as part of the installation process, run License Manager from the **Easysoft** group in the Windows **Start** menu when you're ready to license the Easysoft JDBC-Access Gateway. These types of license are available:

- A free time-limited trial license, which gives you free and unrestricted use of the product for a limited period (usually 14 days).
- A full license if you have purchased the product. On purchasing the product you are given an authorization code, which you use to obtain a license.

To license the Easysoft JDBC-Access Gateway:

1. In License Manager, enter your contact details.

You **must** complete the **Name**, **E-Mail Address**, and **Company** fields.

The e-mail address **must** be the same as the one used to register at the Easysoft web site. Otherwise, you won't be able to obtain a trial license.

2. Choose **Request License**.

You're prompted to choose a license type.

3. Do one of the following:

- For a trial license, choose **Time Limited Trial**, and then choose **Next**.

-Or-

- For a purchased license, choose **Non-expiring License**, and then choose **Next**.

4. Choose your product from the drop-down list when prompted, and then choose **Next**.

5. For a purchased license, enter your authorization code when prompted, and then choose **Next**.

6. Choose how to get your license when prompted.

7. Do one of the following:

- Choose **On-line Request** if your machine is connected to the internet and can make outgoing connections to port 8884.

With this method, License Manager automatically requests and then applies your license.

-Or-

- Choose **View Request**. Then open a web browser and go to https://www.easysoft.com/support/licensing/trial_license.html or https://www.easysoft.com/support/licensing/full_license.html, as appropriate. In the web page, enter your machine number (labelled **Number** in the license request). For purchased licenses, you also need to enter your authorization code (labelled **Ref** in the license request).

We'll automatically email your license to the email address you supplied in License Manager.

-Or-

- Choose **Email Request** to email your license request to our licensing team.

Once we've processed your request, we'll email your license to the email address you supplied in License Manager.

8. Close the License Manager windows and then choose **Finish**.

If you chose either **View Request** or **Email Request**, apply your license by double-clicking the email attachment when you get the license email from us. Alternatively, start License Manager from the **Easysoft** folder in the Windows **Start** menu. Then choose **Enter License** and paste the license in the space provided.

Once you've licensed the Easysoft JDBC-Access Gateway, the installation is complete.

Repairing the installation

The installer can repair a broken Easysoft JDBC-Access Gateway installation. For example, you can use the installer to restore missing Easysoft JDBC-Access Gateway files or registry keys. To do this:

1. In the Windows **Taskbar**, enter Add or remove programs in the Windows **Search** box.
2. Select Easysoft JDBC-Access Gateway in the list, and then choose **Repair**.

Uninstalling on Windows

This section explains how to remove the Easysoft JDBC-Access Gateway from your system.

Removing the Easysoft JDBC-Access Gateway

1. In the Windows **Taskbar**, enter Add or remove programs in the Windows **Search** box.
2. Select Easysoft JDBC-Access Gateway in the list, and then choose **Uninstall**.

Note	Easysoft product licenses are stored in the Windows registry. When you uninstall, your licenses are not removed, so you do not need to relicense the product if you reinstall or upgrade.
-------------	---

Connecting to your Access database

Registering the Easysoft JDBC-Access Gateway

To register the Easysoft JDBC-Access Gateway JDBC driver, your Java application must specify the class `easysoft.sql.esMdbDriver`. For example:

```
Class.forName("easysoft.sql.esMdbDriver");
```

To make this class available, add the Easysoft JDBC-Access Gateway client (`esmdb.jar`) to the system-wide classpath or copy the client to a directory that's already on the system-wide classpath or your application's internal classpath.

The connection URL

When the Easysoft JDBC-Access Gateway JDBC driver is registered, you can establish a connection by using a connection URL and the `getConnection` method of the `DriverManager` class. For example:

```
String connectionUrl = "jdbc:easysoft:mdb?DBQ=C:/Users/Public/Northwind.mdb";
Connection con = DriverManager.getConnection(connectionUrl);
```

To establish a connection with the Easysoft JDBC-Access Gateway, use a connection URL of the form:

```
jdbc:easysoft:mdb?DBQ=<path>[;<odbc-driver-attribute>=<value>]
```

where:

- `<path>` is the path to the Access database (`.mdb`).
- `<odbc-driver-attribute>` is an Access ODBC driver attribute.

You can retrieve the available Access ODBC driver attributes along with a description for each attribute by using the `getPropertyInfo` method of the `Driver` class. For example:

```
public static void driverProperties() {

    // Replace the DBQ value with the path to your Access database.
    String connectionUrl = "jdbc:easysoft:mdb?" +
        "DBQ=C:/Northwind/BlankII.mdb";
    Driver driver = null;
    DriverPropertyInfo props[] = null;

    try {
        Class.forName("easysoft.sql.esMdbDriver");
        driver = DriverManager.getDriver(connectionUrl);
        props = driver.getPropertyInfo(connectionUrl, new Properties());
        System.out.println("JDBC URL Attributes");
        for (int i = 0; i < props.length; i++) {
            System.out.print("\t" + props[i].name);
            System.out.print(" = ");
            System.out.print(props[i].value);
            System.out.print(" : ");
            System.out.println(props[i].description + ".");
        }
    }
}
```

```
    }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Example connection URLs

Opens an Access database and enables Jet 4.0 features:

```
jdbc:easysoft:mdb?DBQ=C:/Users/Public/Northwind.mdb;ExtendedAnsiSQL=True
```

Opens an Access database that's stored in a shared folder:

```
jdbc:easysoft:mdb?DBQ=//mymachine/myshare/Sales.accdb
```

Opens a password-protected Access database from a mapped network drive:

```
jdbc:easysoft:mdb?DBQ=Z:/Orgdata.mdb;PWD=p455w0rd
```

Tracing

This section describes how to enable tracing when using the Easysoft JDBC-Access Gateway.

To help resolve issues and problems with the Easysoft JDBC-Access Gateway, you may need to enable JDBC tracing (also known as logging). This can be a very useful debugging aid, but it should be remembered that tracing will adversely affect performance, and so should be turned off when you have resolved your problem. You turn tracing on by using one of the following methods:

- The `DriverManager.setLogWriter` method lets you specify a `PrintWriter` object that's used to log any JDBC-related information. For example:

```
try {
    Class.forName("easysoft.sql.esMdbDriver");
    DriverManager.setLogWriter(new PrintWriter(System.out));
}
```

- Alternatively, if you use a `DataSource` object to get a connection, you can use the `DataSource.setLogWriter` method to turn on JDBC tracing. For example:

```
try {
    DataSource ds = (DataSource) envCtx.lookup("jdbc/Northwind");
    ds.setLogWriter(new PrintWriter(System.out));
}
```

Note

The Easysoft JDBC-Access Gateway communicates with the Microsoft ODBC driver directly rather than through the Microsoft ODBC Driver Manager. For this reason, the ODBC Driver Manager tracing facility, accessible from the **ODBC Data Source Administrator**, can't be used to log Easysoft JDBC-Access Gateway activity. Error messages returned by the ODBC driver are returned through the JDBC tracing mechanism however. = About the Easysoft JDBC-Access Gateway :description: JRE requirements, data type support, cursor support, and working with Unicode data.

The Easysoft JDBC-Access Gateway provides real-time access to Microsoft Access data from any application that supports JDBC.

- [Java Runtime Environment \(JRE\) requirements](#)
- [Setting the class path](#)
- [Data types](#)
- [Cursors](#)
- [Unicode](#)

Java Runtime Environment (JRE) requirements

The Easysoft JDBC-Access Gateway requires the JRE 1.6.0 or later.

To check whether you have the JRE installed on your machine and that your JRE version is one the Easysoft JDBC-Access Gateway supports, open a Command Prompt window, and enter `java -version`. For example:

```
c:\>java -version
java version "1.7.0"
Java(TM) SE Runtime Environment (build 1.7.0-b147)
Java HotSpot(TM) Client VM (build 21.0-b17, mixed mode, sharing)
```

If the reported JRE version is 1.5.*n* or earlier (or you get the error "'java' is not recognized as an internal or external command, operable program or batch file"), you need to obtain a JRE for your computer.

Windows versions of the JRE are available to download from:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Note

The Java Development Kit (JDK), which includes the JRE, is also available to download from this web page. However, unless you are going to develop a Java application to run against the Easysoft JDBC-Access Gateway, you only need to download the JRE package.

If you have installed a 64-bit version of `aceodbc.dll` (this is the Microsoft driver that you need if you want to connect to ACCDB database files), you need to install a 64-bit JRE. If the output produced by running `java -version` contains 64-Bit Server VM, you have a 64-bit JRE. To check whether `aceodbc.dll` is 64-bit, run **ODBC Data Source Administrator**, which is located in the Windows Control Panel under **Administrative Tools**. Double-click the 64-bit ODBC data sources icon. In the **Drivers** tab, if the list contains Microsoft Access Driver (*.mdb, *.accdb), you have the 64-bit version of `aceodbc.dll`.

If your database is an MDB file, you don't need to install `aceodbc.dll`. The Easysoft JDBC-Access Gateway will use `odbcjt32.dll` to connect to your database. `odbcjt32.dll` is included with the Windows operating system. Because `odbcjt32.dll` is a 32-bit library, you need to use a 32-bit JRE and a 32-bit version of the Easysoft JDBC-Access Gateway. The 32-bit version of the Easysoft JDBC-Access Gateway is installed in:

easysoft-installation-folder\Libs32

The default location for *easysoft-installation-folder* is *drive:\Program Files\Easysoft Limited\Easysoft JDBC-Access Gateway*.

Include *easysoft-installation-folder*\Libs32 in your class path.

Setting the class path

The Easysoft JDBC-Access Gateway Java classes are contained in:

easysoft-installation-folder\Libs\esmdb.jar

The default location for *easysoft-installation-folder* is *drive:\Program Files\Easysoft Limited\Easysoft JDBC-Access Gateway*.

Because esmdb.jar is not part of the Java platform, you need to tell Java where to find the Easysoft JDBC-Access Gateway classes. To do this, you may have to:

- Include *easysoft-installation-folder*\Libs\esmdb.jar in the CLASSPATH environment variable value.
- Include *easysoft-installation-folder*\Libs\esmdb.jar in the java -classpath option value.
- Copy esmdb.jar to a folder that is reserved for third party .jar files by your application.

Data types

The Easysoft JDBC-Access Gateway supports the following Jet SQL data types and data type synonyms:

Data type	Synonym
BIT	LOGICAL LOGICAL1 YESNO
TINYINT	INTEGER1 BYTE
MONEY	CURRENCY
DATETIME	DATE TIME
REAL	SINGLE FLOAT4 IEEESINGLE
FLOAT	DOUBLE FLOAT8 IEEEDOUBLE NUMBER
SMALLINT	SHORT INTEGER2
INTEGER	LONG INT INTEGER4
DECIMAL	NUMERIC DEC
TEXT	LONGTEXT LONGCHAR MEMO NOTE NTEXT

Data type	Synonym
CHAR	CHARACTER STRING
	VARCHAR
	CHARACTER VARYING
	NCHAR
	NATIONAL CHARACTER
	NATIONAL CHAR
	NATIONAL CHARACTER VARYING
	NATIONAL CHAR VARYING

Notes

- In the Microsoft Access interface, a **Table Design View** field type may have a different name to the Jet SQL data type it corresponds with. For example, the **Table Design View** field type for the MONEY data type is CURRENCY.
- To create columns with the data type TINYINT or DECIMAL, or the synonym DEC, NTEXT, CHARACTER, CHARACTER VARYING, NCHAR, NATIONAL CHARACTER, NATIONAL CHAR, NATIONAL CHARACTER VARYING, or NATIONAL CHAR VARYING, you need to set the ExtendedAnsiSQL Access ODBC driver option to 1. For example:

```
String connectionString =
"jdbc:easysoft:mdb?DBQ=C:/Users/Public/Northwind.mdb;ExtendedAnsiSQL=1";
```

Otherwise, your CREATE TABLE or ALTER TABLE statement will fail with the error "[Microsoft][ODBC Microsoft Access Driver] Syntax error in field definition."

- You cannot use literal GUIDs in SQL queries. Literal GUIDs are needed to filter records by values contained in UNIQUEIDENTIFIER or GUID columns. The Access ODBC driver does not support the Access SQL syntax for specifying GUID literals ({GUID {nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnn}}). For example, this query:

```
SELECT * FROM MyTable WHERE MyUniqueIdentifierCol = {guid {019CE4C0-D57C-68A6-
0000-000000000109}};
```

fails with the error:

```
[Microsoft][ODBC Microsoft Access Driver] Syntax error (missing operator) in
query expression
```

Cursors

The set of rows returned by a SQL query consists of all the rows that satisfy the conditions of that query, and is known as the result set. Applications can't always work effectively with the entire result set as a unit. These applications need a mechanism to work with one row or a small block of rows at a time. Cursors are an extension to result sets that provide that mechanism.

The Easysoft JDBC-Access Gateway supports the following cursor types:

Cursor type	Description
TYPE_FORWARD_ONLY	The result set is not scrollable. The cursor moves only forward, from before the first row to after the last row. This is the default behaviour for a result set.
TYPE_SCROLL_INSENSITIVE	The result set is scrollable but not sensitive to changes to the data that underlies the result set; its cursor can move both forward and backward relative to the current position, and it can move to an absolute position.

TYPE_FORWARD_ONLY Example

```
try {

    // Establish the connection.
    Class.forName("easysoft.sql.esMdbDriver");
    Connection con =
    DriverManager.getConnection("jdbc:easysoft:mdb?DBQ=C:/Users/Public/Northwind.mdb");

    // Use a forward only cursor to move forwards through a result
    // set. This is the default cursor type, and so the following
    // line could be replaced with
    // Statement stmt = con.createStatement()
    Statement stmt = con.createStatement(ResultSet.TYPE_FORWARD_ONLY,
    ResultSet.CONCUR_READ_ONLY);
    ResultSet rs = stmt.executeQuery("SELECT SupplierID, CompanyName FROM
    Suppliers");

    // Move forwards through the result set.
    while (rs.next()) {
        int id = rs.getInt("SupplierID");
        String supplier = rs.getString("CompanyName");
        System.out.println(id + "\t" + supplier);
    }
}

// Handle any errors that may have occurred.
catch (Exception e) {
    e.printStackTrace();
}
```


TYPE_SCROLL_INSENSITIVE Example

```
try {
    // Establish the connection.
    Class.forName("easysoft.sql.esMdbDriver");
    Connection con =
DriverManager.getConnection("jdbc:easysoft:mdb?DBQ=C:/Users/Public/Northwind.mdb");

    // Use a scrollable cursor to move backwards through a result
    // set.
    Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
    ResultSet srs = stmt.executeQuery("SELECT SupplierID, CompanyName FROM
Suppliers");

    // Move the cursor to the last record in the result set.
    srs.last();
    int id = srs.getInt("SupplierID");
    String supplier = srs.getString("CompanyName");
    System.out.println(id + "\t" + supplier);

    // Move backwards through the remaining records.
    while (srs.previous()) {
        id = srs.getInt("SupplierID");
        supplier = srs.getString("CompanyName");
        System.out.println(id + "\t" + supplier);
    }
}

// Handle any errors that may have occurred.
catch (Exception e) {
    e.printStackTrace();
}
```

Unicode

Background

Unicode is a computing industry standard designed to consistently and uniquely encode characters used in written languages throughout the world.

Java's native character encoding is Unicode. The primitive type `char` is a single Unicode character. The `String` class is a collection of characters.

The JDBC data types used to express character data, (`CHAR`, `VARCHAR`, and `LONGVARCHAR`) can all be represented as a Java `String`. (The Easysoft JDBC-Access Gateway maps the Access `TEXT` and `CHAR` data types to the `VARCHAR` and `CHAR` JDBC data types.)

Unicode support was added in Jet version 4.0. A Jet version 4.0 `.mdb` file uses the Unicode encoding to store character data in `TEXT` and `CHAR` columns.

The Access ODBC driver supports Unicode in the form of Unicode data types and Unicode versions of the ODBC API.

Unicode and the Easysoft JDBC-Access Gateway

The Easysoft JDBC-Access Gateway allows Unicode data to be specified in:

- The JDBC connection URL. For example:

```
String connectionUrl =
"jdbc:easysoft:mdb?DBQ=C:/Users/Public/UnicodeCharsDB.mdb";
Connection con = DriverManager.getConnection(connectionUrl);
```

- SQL statements. For example:

```
Connection con = null;
Statement stmt = null;
ResultSet rs = null;
try {
    Class.forName("easysoft.sql.esMdbDriver");
    con = DriverManager.getConnection(connectionUrl);

    String SQL = "INSERT INTOUnicodeCharsTable(UnicodeCharsColumn) VALUES
('UnicodeChars')";
    stmt = con.createStatement();
    stmt.execute(SQL);

    SQL = "SELECTUnicodeCharsColumn FROMUnicodeCharsTable";
    rs = stmt.executeQuery(SQL);
    while (rs.next()) {
        System.out.println(rs.getString(1));
    }
}
```

- Metadata (table names and so on). For example:

```
Connection con = null;
DatabaseMetaData dm = null;
```

```

ResultSet rs = null;

try {
    Class.forName("easysoft.sql.esMdbDriver");
    con = DriverManager.getConnection(connectionUrl);
    dm = con.getMetaData();
    rs = dm.getColumns(null, null, "UnicodeCharsTable", null);
    System.out.println("Columns");
    while (rs.next()) {
        System.out.println("\t" + rs.getString("COLUMN_NAME") +
            " : " + rs.getString("TYPE_NAME"));
    }
}

```

- SQL statement parameters in parameterized queries. For example:

```

Connection con = null;
PreparedStatement pstmt = null;
ResultSet rs = null;
ResultSetMetaData rsmd = null;

try {
    Class.forName("easysoft.sql.esMdbDriver");
    con = DriverManager.getConnection(connectionUrl);
    String SQL = "SELECT * FROMUnicodeCharsTable whereUnicodeCharsColumn = ?";
    pstmt = con.prepareStatement(SQL);
    pstmt.setString(1, "UnicodeChars");
    rs = pstmt.executeQuery();
    rsmd = rs.getMetaData();
    while (rs.next()) {
        for (int i = 1; i <= rsmd.getColumnCount(); i++) {
            System.out.print(rs.getString(i) + "\t");
        }
        System.out.println();
    }
}

```

Note

If you experience text corruption when working with Unicode data (for example, you get a ? character instead of the expected character), you may need to explicitly specify a Unicode encoding in your client application. For example, when testing the Easysoft JDBC-Access Gateway with Eclipse and Apache Tomcat we need to set the character encoding to UTF-8. (In Eclipse, we set the Text file encoding to UTF-8; in Tomcat we added this line to our test JSP: `<%@page contentType="text/html; charset=UTF-8"%>`.)

Index

C

- class path, [13](#)
- connecting to Access, [9](#)
- connection URL, [9](#)
- cursor support, [16](#)

D

- data types, [14](#)

E

- Easysoft JDBC-Access Gateway
 - connecting to Access, [9](#)
 - connection URL, [9](#)
 - cursor support, [16](#)
 - installing
 - on Windows, [6](#)
 - overview, [5](#)
 - setting the class path, [13](#)
 - supported data types, [14](#)
 - tracing, [11](#)
 - Unicode support, [18](#)
 - uninstalling
 - on Windows, [8](#)

J

- JRE requirements, [12](#)

L

- logging, [11](#)

T

- tracing, [11](#)

U

- Unicode, [18](#)
- uninstalling
 - on Windows, [8](#)