

# Welcome to DBExtractor!

Copyright © 2005-2006 SQLExec LLC. All rights reserved.

[DBExtractor Home Page](#)  
[Technical Support](#)

**DBExtractor** is an ODBC database export utility. It gathers execution parameters from the **DBExtractor Specifications File**, which is then used to connect to specific databases via an ODBC driver and execute user-defined queries or special [metadata queries](#). It is designed to work as a data refresh utility (see Date Driver) invoked as a windows [scheduled task](#), where new or updated data is found and extracted to disk. This program runs on all Windows operating systems, and can work with any ODBC-compliant database driver. DBExtractor is licensed, not sold. See the [license agreement](#) for details.

## Footprint

**DBExtractor** has a relatively small application footprint. It takes up very little space in the computer's memory. It uses a minimum of disk space. CPU utilization is also minimal. File I/O may be intensive at times depending upon the amounts of data being extracted from a database, and output file sizes also vary in this respect depending upon the number of rows being returned from database query calls.

## Auditing/Reporting/Logging

**DBExtractor** also has a robust [logging facility](#), where informational, warnings, and errors are recorded to the program log file. There is no auditing feature per se, but the log file does keep track of every SQL statement that was executed and how many rows were returned.

## Security

**DBExtractor** has [security features](#) involving the capacity to encrypt passwords used when connecting to a database. **DBExtractor** requires only a READ ACCESS to tables from which it will execute queries.

## Date Driver

[Date Driver](#) refers to the logic employed to get new or updated data from the source database. It involves updating the **DBExtractor Specifications File** with the current system date, but only when the program runs completely without errors. This date is then used as the start point for successive executions of this program. The current system date is used as the end point. SQL queries defined in this input file have special keywords designating start and end date where they are replaced with real-time values at execution time.

## SETUP Procedures

The [setup](#) procedures are straightforward to get the program up and running as a scheduled DOS batch job.

## Execution Parameters

**DBExtractor** takes only one optional parameter, the location of the **DBExtractor Specifications File**. If not provided, it will look for a specific file, **DBEXTRACTOR.INI** in the DBExtractor program directory.

# License Agreement

## **END-USER LICENSE AGREEMENT FOR THE SOFTWARE PRODUCT known as DBExtractor**

IMPORTANT - READ CAREFULLY: This DBExtractor End-User License Agreement ("EULA") is a legal agreement between you ((to include affiliates and subsidiaries), hereby referred to as the LICENSEE, and SQLEXEC LLC, the PRODUCT OWNER of the SOFTWARE PRODUCT identified above and hereafter referred to as the LICENSOR. The specific type of LICENSE (unrestricted site, restricted site, seat, etc.) is based on the INVOICE or CONTRACT that was delivered to the LICENSEE along with the SOFTWARE PRODUCT. This SOFTWARE PRODUCT runs on a COMPUTER, which consists of computers, workstations, terminals, or other electronic or digital devices that run on the operating systems noted in this license agreement.

The SOFTWARE PRODUCT consists of all files and software distributed with the SOFTWARE PRODUCT. These components are distributed as part of this license agreement by the LICENSOR to the LICENSEE. The source code used to compile or produce the binary output files are not distributed as part of this license agreement, and in no way are any rights to this source code implied by this license agreement to the LICENSEE. Only the LICENSOR can provide maintenance of this SOFTWARE PRODUCT.

The LICENSOR guarantees the product to be free of defects, known or unknown at the time of this LICENSE agreement, for the duration of the LICENSE agreement. The LICENSOR will, at the LICENSOR's expense, fix any defect brought to its attention by the LICENSEE in a timely manner befitting the nature of the defect and the business impact on the LICENSEE. Any updates to the SOFTWARE PRODUCT by the LICENSOR as a result of fixing defects, are also covered by all provisions of this license agreement.

By installing, copying, downloading, accessing, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or use the SOFTWARE PRODUCT.

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

### **SOFTWARE PRODUCT LIMITATIONS**

1. The SOFTWARE PRODUCT license only applies to the following Windows operating systems:  
Windows NT 4  
Windows 2000 (all versions)  
Windows XP (all versions)
2. The SOFTWARE PRODUCT is guaranteed to be ODBC-compliant and only works with ODBC drivers. As such it is guaranteed to work without defect as long as the ODBC driver required by this SOFTWARE PRODUCT is without ODBC defects in any way used by this SOFTWARE PRODUCT.
3. The SOFTWARE PRODUCT does not work with BLOB/CLOB or other binary database formats.

Reservation of Rights. All rights not expressly granted are reserved by the

LICENSOR of the SOFTWARE PRODUCT.

**GRANT OF LICENSE. This EULA grants you the following rights:**

Applications Software. The LICENSEE may make additional copies of the SOFTWARE PRODUCT for backup purposes only.

Unlimited instances of this SOFTWARE PRODUCT may run or be executed on a single COMPUTER, where this license is in effect.

**VALID LICENSE TYPES**

The following forms of license agreements are associated with this SOFTWARE PRODUCT. The INVOICE for this SOFTWARE PRODUCT contains information about what type of license is granted to the LICENSEE of this SOFTWARE PRODUCT. Here are the current license types available:

**Unrestricted Site License:**

An unrestricted site license allows any user within your company to use the SOFTWARE PRODUCT or any number of instances of the SOFTWARE PRODUCT on any workstation that they prefer. Unrestricted Site Licenses are limited in duration and must be renewed according to the terms of the purchase agreement. **Unrestricted Site Licenses do not convey or imply a Corporate Distribution License (CDL). A CDL is where the LICENSEE may combine a SOFTWARE PRODUCT with other products not part of this license agreement or distribute this SOFTWARE PRODUCT as a sublicense to others.**

**Restricted Site License:**

A Restricted site license allows a limited number of users within your company to use the SOFTWARE PRODUCT concurrently on any COMPUTER within that company. Restricted Site Licenses are limited in duration and must be renewed according to the terms of the purchase agreement. **Restricted Site Licenses do not convey or imply a Corporate Distribution License (CDL). A CDL is where the LICENSEE may combine a SOFTWARE PRODUCT with other products not part of this license agreement or distribute this SOFTWARE PRODUCT as a sublicense to others.**

**Seat License:**

Seat licenses grants one concurrent user to interact with the SOFTWARE PRODUCT on a limited number of COMPUTERS that are also owned by the LICENSEE. This limited number of COMPUTERS can be no more that three additional COMPUTERS. There is no time limit on a seat license agreement so the LICENSEE is not required to renew this license agreement. The seat license grants free upgrades to the LICENSEE for a limited period of time. This period is stipulated in the INVOICE agreement entered into by the LICENSEE. This period is not less than one year from the receipt of this SOFTWARE PRODUCT.

\* Limitations on Decompilation, and Disassembly. You may not decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

\* Trademarks. This EULA does not grant you any rights in connection with any trademarks or service marks of SQLEXEC, or the SOFTWARE PRODUCT.

\* Rental. You may not rent, lease, or sell the SOFTWARE PRODUCT as a stand-alone product.

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

EXPORT RESTRICTIONS. You agree that you will not export or re-export the SOFTWARE PRODUCT, any part thereof, or any process or service that is the

direct product of the SOFTWARE PRODUCT to any country, person or entity subject to U.S. export restrictions. You specifically agree not to export or re-export any of the Restricted Components (i) to any country to which the U.S. has embargoed or restricted the export of goods or services, which currently include, but are not necessarily limited to Cuba, Iran, Iraq, Libya, North Korea, Sudan and Syria, or to any national of any such country, wherever located, who intends to transmit or transport the Restricted Components back to such country; (ii) to any person or entity who you know or have reason to know will utilize the Restricted Components in the design, development or production of nuclear, chemical or biological weapons; or (iii) to any person or entity who has been prohibited from participating in U.S. export transactions by any federal agency of the U.S. government. You warrant and represent that neither the BXA nor any other U.S. federal agency has suspended, revoked or denied your export privileges.

If you acquired this SOFTWARE PRODUCT in the United States, this EULA is governed by the laws of the State of Virginia. If this SOFTWARE PRODUCT was acquired outside the United States, then local law may apply.

LIMITATION OF LIABILITY. To the maximum extent permitted by applicable law, in no event shall the LICENSOR or its suppliers be liable for any special, incidental, indirect, or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use the SOFTWARE PRODUCT or the provision of or failure to provide Support Services, even if the LICENSOR has been advised of the possibility of such damages. In any case, the LICENSOR's entire liability under any provision of this EULA shall be limited to the greater of the amount actually paid by you for the SOFTWARE PRODUCT or U.S.\$5.00; provided, however, if you have entered into a SOFTWARE PRODUCT Support Services Agreement, the LICENSOR's entire liability regarding Support Services shall be governed by the terms of that agreement. Because some states and jurisdictions do not allow the exclusion or limitation of liability, the above limitation may not apply to you.

# Date Driver

The basic purpose of the **date driver** logic is to get data that is new or changed since the last run. It could also be used to get all of the original data on the first run for a particular site. In that case, you would not use BEGIN\_DATE or END\_DATE keywords. The DBExtractor program does not validate that you are using the keywords or using them correctly. It just does global replacements for them if found. It replaces the BEGIN\_DATE with the LASTCOMMITDATE and the END\_DATE with the local, system date.

The SQL statements must be coded so that the queried date is greater than or equal to the BEGIN\_DATE and less than the END\_DATE. The LASTCOMMIT DATE is only updated if the previous DBExtractor execution completely successfully without any fatal errors.

The format of the LASTCOMMITDATE determines how the interim period is defined. For the Date format, updates are found based on a day units, whereas updates are found for time units using the DateTime format.

## Date Format: YYYYMMDD, YYYY/MM/DD, or YYYY-MM-DD

This interim period are the **days** greater than or equal to the last commit date (LASTCOMMITDATE value in the DBExtract Specifications File) and less than the local, system date.

## DateTime Format: YYYYMMDDHHMMSS

This interim period is the **time** greater than or equal to the last commit datetime (LASTCOMMITDATE value in the DBExtractor Specifications File) and less than the current system datetime.

The SQL statements provided by the user must have the placeholders for start and end date embedded in the SQL statements. These placeholders for start and end date are **%%BEGIN\_DATE%%** and **%%END\_DATE%%** respectively. They must be enclosed in quotes since they are treated as constant character strings within the SQL statement. Consider the following example in Oracle:

```
select * from linxdata.temp_ccso_arrest w here TO_CHAR(addtime, 'YYYYMMDD') >= '%%BEGIN_DATE%%' and TO_CHAR(addtime, 'YYYYMMDD') < '%%END_DATE%%';
```

Below are some sample functions to use for these date predicates based on certain database vendors. These examples use the DATE format.

### Oracle

```
TO_CHAR(<date field>, 'YYYYMMDD') >= '%%BEGIN_DATE%%' and TO_CHAR(<date field>, 'YYYYMMDD') < '%%END_DATE%%'
```

### MS SQL Server

```
convert(char(8), <date field>, 112) >= '%%BEGIN_DATE%%' and convert(char(8), <date field>, 112) < '%%END_DATE%%'  
CAST(FLOOR(CAST(<date field> AS float)) AS datetime) >= '%%BEGIN_DATE%%' and CAST(FLOOR(CAST(<date field> AS float)) AS datetime) < '%%END_DATE%%'
```

### Adaptive Server Anywhere (Sybase ASA or SQLAnywhere)

```
convert(char(8), <date field>, 112) >= '%%BEGIN_DATE%%' and convert(char(8), <date field>, 112) < '%%END_DATE%%'
```

### Adaptive Server Enterprise (Sybase ASE or Sybase SQL Server)

```
convert(char(8), <date field>, 112) >= '%%BEGIN_DATE%%' and convert(char(8), <date field>, 112) < '%%END_DATE%%'
```

### MS ACCESS

```
Format(<date field>, 'yyyymmdd') >= '%%BEGIN_DATE%%' and Format(<date field>, 'yyyymmdd') < '%%END_DATE%%'
```

## INTERBASE/FIREBIRD

WHERE <date field> >= CAST( '%%BEGIN\_DATE%%' AS DATE) AND <date field> < CAST('%%END\_DATE%%' AS DATE)

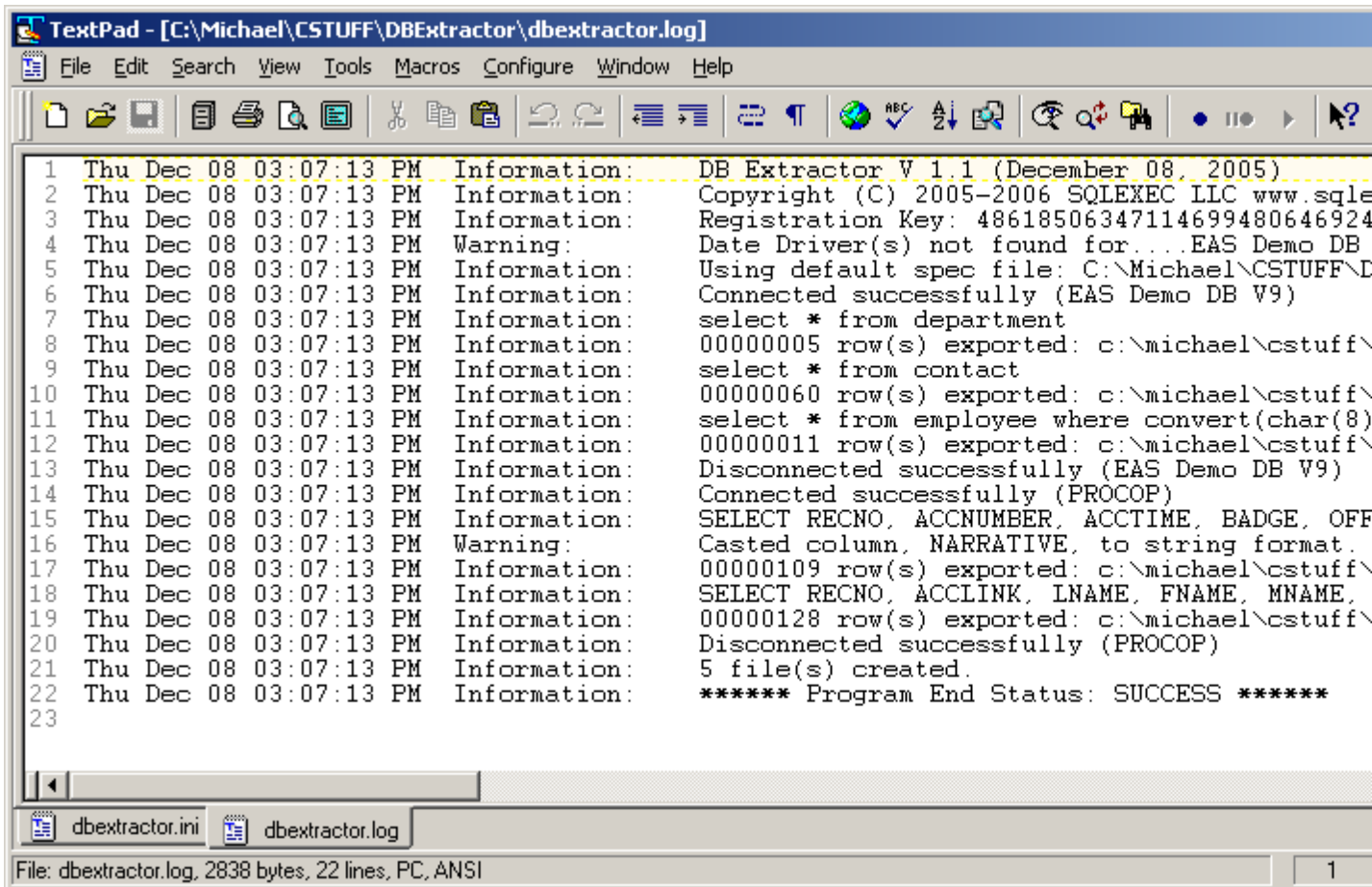
# Logging

**DBExtractor** writes error and logging information to the file specified as the **DBExtractor** log file. By default, this log file is **DBExtractor.log** in the **DBExtractor** program directory if not specified otherwise on the **LOGFILE** parameter in the **DBExtract Specifications File**.

**DBExtractor** provides a robust logging feature. You can specify the location of the log file in the **DBExtractor Specifications File**, or it defaults to **DBExtractor.log** in the **DBExtractor** program directory. Records are appended to the logging file. The **DBExtractor** logging file contains a timestamp for each record entry. A record entry record is identified by one of the following severity codes:

- DEBUG
- INFORMATION
- WARNING
- NON FATAL ERROR
- FATAL ERROR

A text string follows the severity code with a detailed description of the **DBExtractor** logging entry. The log file is automatically deleted when the file size exceeds 500Kbytes.



```
TextPad - [C:\Michael\CSTUFF\DBExtractor\dbextractor.log]
File Edit Search View Tools Macros Configure Window Help
1 Thu Dec 08 03:07:13 PM Information: DB Extractor V 1.1 (December 08, 2005)
2 Thu Dec 08 03:07:13 PM Information: Copyright (C) 2005-2006 SQLEXEC LLC www.sql
3 Thu Dec 08 03:07:13 PM Information: Registration Key: 48618506347114699480646924
4 Thu Dec 08 03:07:13 PM Warning: Date Driver(s) not found for...EAS Demo DB
5 Thu Dec 08 03:07:13 PM Information: Using default spec file: C:\Michael\CSTUFF\D
6 Thu Dec 08 03:07:13 PM Information: Connected successfully (EAS Demo DB V9)
7 Thu Dec 08 03:07:13 PM Information: select * from department
8 Thu Dec 08 03:07:13 PM Information: 00000005 row(s) exported: c:\michael\cstuff\
9 Thu Dec 08 03:07:13 PM Information: select * from contact
10 Thu Dec 08 03:07:13 PM Information: 00000060 row(s) exported: c:\michael\cstuff\
11 Thu Dec 08 03:07:13 PM Information: select * from employee where convert(char(8)
12 Thu Dec 08 03:07:13 PM Information: 00000011 row(s) exported: c:\michael\cstuff\
13 Thu Dec 08 03:07:13 PM Information: Disconnected successfully (EAS Demo DB V9)
14 Thu Dec 08 03:07:13 PM Information: Connected successfully (PROCOP)
15 Thu Dec 08 03:07:13 PM Information: SELECT RECNO, ACCNUMBER, ACCTIME, BADGE, OFF
16 Thu Dec 08 03:07:13 PM Warning: Casted column, NARRATIVE, to string format.
17 Thu Dec 08 03:07:13 PM Information: 00000109 row(s) exported: c:\michael\cstuff\
18 Thu Dec 08 03:07:13 PM Information: SELECT RECNO, ACCLINK, LNAME, FNAME, MNAME,
19 Thu Dec 08 03:07:13 PM Information: 00000128 row(s) exported: c:\michael\cstuff\
20 Thu Dec 08 03:07:13 PM Information: Disconnected successfully (PROCOP)
21 Thu Dec 08 03:07:13 PM Information: 5 file(s) created.
22 Thu Dec 08 03:07:13 PM Information: ***** Program End Status: SUCCESS *****
23
File: dbextractor.log, 2838 bytes, 22 lines, PC, ANSI
```

# Oracle SQL Loader Considerations

This section documents some notes related to the loading of data using Oracle's SQL Loader that had been previously extracted using **DBExtractor**.

**DBExtractor** does not place double-quotes around output fields, so you must remove the following phrase if found in your Oracle SQL Loader control file:

**optionally enclosed by ""**

This phrase is usually found in control files to detect string fields and to avoid problems with embedded delimiters. **DBExtractor** detects and reports embedded delimiters in the output log file as a fatal error message.

## SQL Loader Peculiarities:

1. For CLOB columns, you must specify **CHAR (40000)**, or you may get data field exceeded max limit errors.
2. You can qualify date column values with a string mask in one of the following ways:  
<column name> DATE 'YYYY/MM/DD HH24:MI:SS',  
<column name> "to\_date(:<column name>, 'YYYY-MM-DD HH24:MI:SS')",



## Metadata Queries

**DBExtractor** can execute queries which return database metadata information. This information is also referred to as the database system tables, where information about tables, columns, column attributes, indexes, and primary/foreign keys are stored. These queries might be useful in gathering information that will help you construct the SQL statements to retrieve the data you want to extract.

Key phrases are used instead of actual SQL statements after the keyword, **SQL=**.

```
SQL=SYSTEMTABLES;FILE=SYSTEMTABLES.DAT
SQL=USERTABLES;FILE=USERTABLES.DAT
SQL=COLUMNS:mytablename;FILE=USERTABLES.DAT
SQL=TABLESTATS:mytablename;FILE=TABLE.DAT
SQL=PKEY:TABLE1;FILE=TABLE1PKEY.DAT
SQL=FKEYS:TABLE1;FILE=TABLE1FKEYS.DAT
```

**SYSTEMTABLES** returns a comma-delimited list of all system tables in the database.

**USERTABLES** returns a comma-delimited list of all user tables in the database.

**COLUMNS** returns a tab-delimited list of all column names for a particular table.

**TABLESTATS** returns index information for a particular table.

**PKEY** returns a tab-delimited list of primary key information.

**FKEYS** returns a tab-delimited list of foreign key information where the table named has foreign keys and where other tables have foreign keys referencing the named table.

# Setup

**DBExtractor** setup procedures involve the following steps:

1. Defining **DBExtractor** query rights on the database host or server.
2. Installing an ODBC driver, and then defining a local Data Source Name for that driver.
3. Entering parameters in the **DBExtractor Specifications File**.
4. Setting up **DBExtractor** to run as a scheduled batch job.

Optionally:

5. Define **DBExtractor** Security, which involves encryption of passwords.

## Database Server

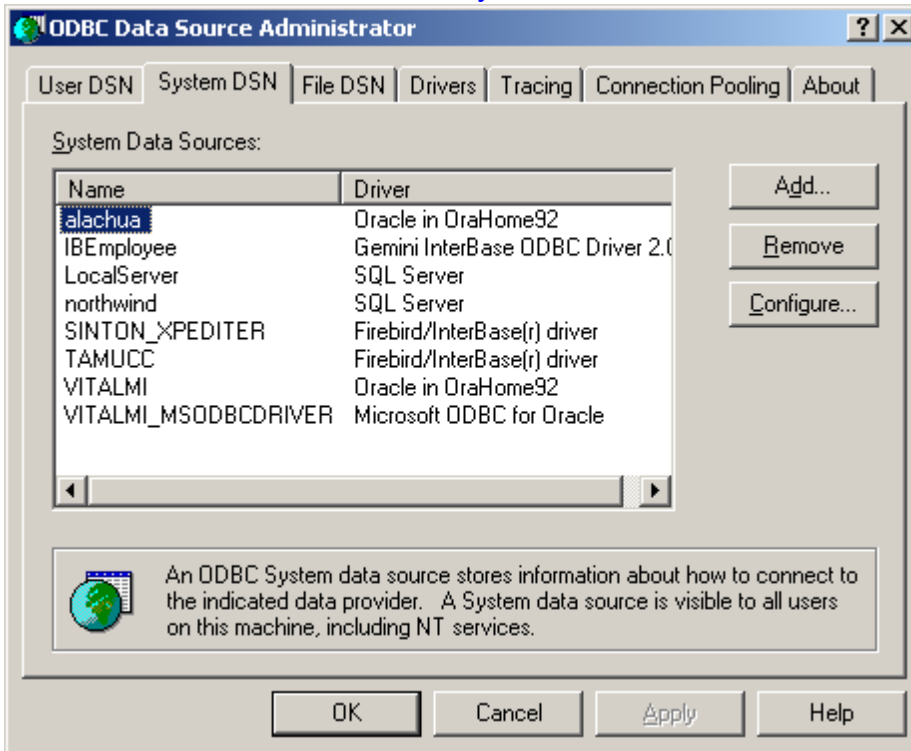
**DBExtractor** needs a valid database profile (userid and password) with which to connect to the database server. The minimal **DBExtractor** rights necessary are SELECT authority with respect to certain tables.

# ODBC

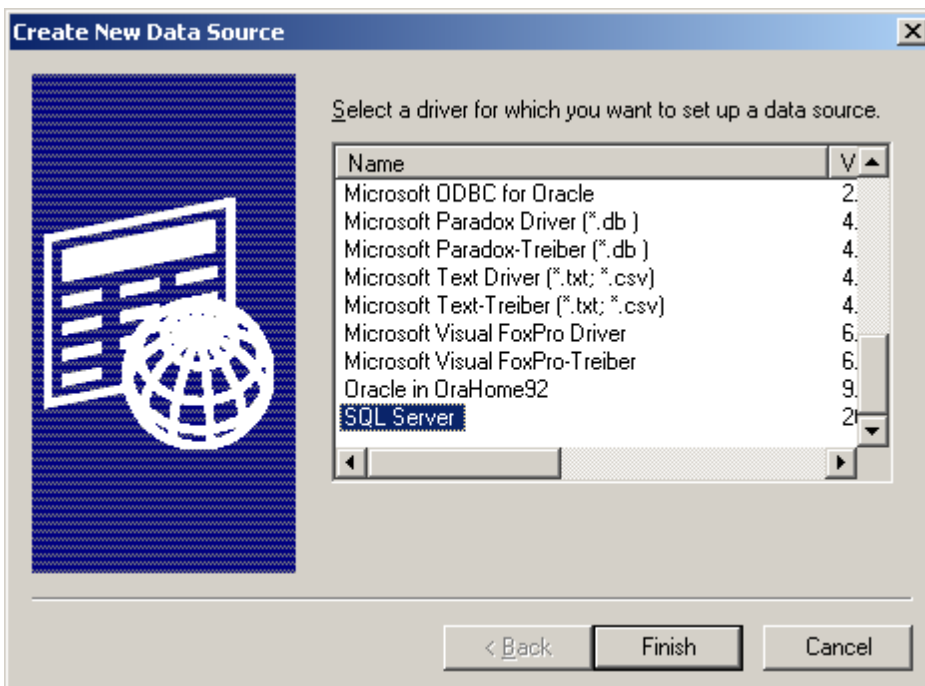
**DBExtractor** requires that an ODBC driver is installed and a Data Source Name (DSN) is created under the **Microsoft ODBC/Data Source Administrator** interface that is used to map a DSN to a specific ODBC driver. The DSN name is then put into the DBExtractor.ini file to identify a database profile to use during program execution.

To add a DSN, follow these steps:

Start the ODBC Administrator, Data Sources (ODBC), which is located under Administrative Tools. Select the System tab, and click ADD.



Select the driver you want to use for the target database vendor. Then press FINISH.



The rest of the steps necessary for defining the DSN are based on the particular interface the ODBC driver provides.

# Specification File

The Specifications file is used to drive the **DBExtractor** program. A sample one is shown below. This section documents the details of each field in this file.

2 forward slashes indicate a comment line which is ignored when the program reads from this file.

The file consists of 2 sections: one general input section and one or more database-specific sections.

The field keywords are case-sensitive and all are UPPERCASE. Values can be either upper or lower case.

## General Parameters

### PRODUCTKEY

This field contains a valid product key if the product was registered successfully. Otherwise, the program runs in trial mode and is limited in functionality (only 50 rows can be retrieved from a table).

### DELIMITER

This field dictates what delimiter will be used to separate the data elements for each row in the resulting output file. You can use 0-4 characters to designate the delimiter. To designate a tab, simple tab over from the equals sign.

### FIXEDWIDTH

Valid values are Y or N. If Y, then each data type field will be padded to the maximum length, otherwise fields are written to the output file as returned from the database driver. For certain drivers, character strings are returned as the maximum length with spaces appended to the end. Some other drivers return the same type of data type with the values trimmed on the right. If you want to make sure all spaces are removed from the end of strings, then set the TRIM parameter to Y. This will enable the program to override the default values returned from the database for string type data fields. You can then keep the fixedwidth settings for non-string type fields separate from how you treat string type fields when you use both the FIXEDWIDTH=N and TRIM=Y parameters. See the table below under TRIM description.

### TRIM

Valid values are Y or N. If Y, then all character fields are trimmed left and right. If you want to make sure all spaces are removed from the end of strings, then set this parameter to Y. See the table below for how FIXEDWIDTH and TRIM work together.

FW	TR	Result
==	==	=====
Y	Y	Invalid. You cannot specify Y for both FIXEDWIDTH and TRIM.
Y	F	Fixed width output for all data type fields.
F	T	All field values are returned as their actual value length In addition, strings are left and right trimmed.
F	F	All field values are returned as their actual value length and strings are not trimmed if they have leading/trailing spaces.

### HEADER

Valid values are Y or N. If Y, then a header row with the column names will be at the top of the output file.

### CR

You can use zero characters to indicate not to replace the carriage returns or any 2 character string length value to specify the replacement values for all carriage returns found in text columns.

### ORACLECTL

Valid values are Y or N. If Y, then Oracle SQL Loader control files are created for every SQL statement. CTL

files parse fields as delimited or fixed width. If FIXEDWIDTH=Y, then CTL files use the fixed width format. Otherwise they use the DELIMITER value for parsing the fields in the output control files.

### **VERBOSE**

Valid values are Y or N. If Y, then additional messages are logged to the standout output terminal. This field does not affect what records are written to the log file.

### **DEBUG**

Valid values are Y or N. If Y, then debugging information is also shown and logged.

### **LOGFILE**

If specified, the path and name of the file where program messages are appended. If empty the logfile name defaults to DBEXTRACTOR.LOG in the program directory.

### **PASSWORDSENCRYPTED**

Valid values are Y or N. If Y, then the password associated with each DSN defined are encrypted and need to be decrypted.

### **EXTRACTDIR**

The name of the local directory where the output files are extracted. Make sure this directory contains nothing else besides files to be extracted, since this directory is flushed before every program execution.

### **EXTRACTDIRCHECK**

Valid values are Y or N. If Y, then any EXE or BAT file is found in the local directory, the program will end with a fatal error. This check helps to ensure that the user does not accidentally delete files in a wrongly specified extract directory. All files are deleted from the extract directory when the DBExtractor program executes.

**NOTE: Currently, this value is always marked as Y regardless of the setting.**

### **DSNCOUNT**

Specifies the number of DSNs that are defined below in the DSN sections. They must match or an error is returned.

### **DATEMASK**

Specifies the LASTCOMMITDATE format for dates. This value is ignored for timestamp values.

Valid values are YYYYMMDD, YYYY/MM/DD, or YYYY-MM-DD. The DATEMASK and LASTCOMMITDATE formats must match.

DATEMASK=YYMMDD

LASTCOMMITDATE=20050115

### **LASTCOMMITDATE**

The date when the last program ran successfully without any fatal errors. The format of the date is defined in

**DATEMASK**. For timestamp values the DATEMASK is ignored. The LASTCOMMITDATE field is used to determine the start date of subsequent program executions. The DATEMASK and LASTCOMMITDATE formats must match.

DATEMASK=YY/MM/DD

LASTCOMMITDATE=2005/01/15

## **Database Profile Section**

This section documents the database(s) to use for extracting the data. You can have one or more DSN sections as shown below. Currently, the maximum number of DSNs that can be specified is 20.

### **[DSN=<DSN Name>]**

The name of the DSN that dictates what ODBC driver to use. This field must be enclosed in brackets as shown above.

### **UID**

The user id used for connecting to the database.

**PWD**

The password used for connecting to the database. This value may be encrypted depending on the value of the general parameter, PASSWORDSENCRYPTED.

**SRV**

The server name of the database. This field usually remains empty.

One or more lines like the following line that designate what SQL to run and to what output file. An SQL statement cannot be defined across multiple lines.

**SQL=** <sql statement>;**FILE=**<file name>

<sql statement> can consist of global replace keywords shown below that designate the start date and end date of the data to retrieve from the database. These keywords must be enclosed in single quotes.

%%**BEGIN\_DATE**%%

%%**END\_DATE**%%

See [Date Driver](#) for more details



# Specification File Sample

[GENERAL PARAMETERS]

PRODUCTKEY=  
DELIMITER=|\*  
FIXEDWIDTH=N  
TRIM=Y  
HEADER=N  
CR=^^  
ORACLECTL=Y  
VERBOSE=Y  
DEBUG=N  
LOGFILE=c:\temp\dbextractor.log  
PASSWORDSENCRYPTED=N  
EXTRACTDIR=c:\temp\files  
EXTRACTDIRCHECK=Y  
DSNCOUNT=3  
DATEMASK=YYYY/MM/DD  
LASTCOMMITDATE=1997/01/01

[DSN=EAS Demo DB V9]

UID=dba  
PWD=sql  
SRV=  
SQL=SELECT \* from w ebdata;FILE=WEBDATA.DAT  
SQL=select \* from department;FILE=DEPARTMENT.DAT  
SQL=select \* from contact;FILE=CONTACT.DAT  
SQL=select \* from employee where convert(char(8), start\_date, 112) >= '%%BEGIN\_DATE%%' and convert(char(8), start\_date, 112) < '%%END\_DATE%%';FILE=EMPLOYEE.DAT

[DSN=TAMUCC]

UID=SYSDBA  
PWD=masterkey  
SRV=  
SQL=select modified from citations;FILE=CITATIONS.DAT  
SQL=SELECT CIT\_ID, TICKETNUM, CTYPE, WARNING, DOI, OFC\_ID, BADGE, OFC, LOCATION, DIRECTION, ADDRESS, DESCRIPTION, TRESPASS, TRESPASS\_EXP, SEARCHED, CONSENTED, NAME\_ID, NAME, NADDRESS, NCITY, NSTATE, NZIP, NPHONE, NHEIGHT, NWEIGHT, NHAIR, NEYES, NRACE, NETHNICITY, NGENDER, NDOB, NSSN, NLICSTATE, NLICNUMBER, NLICTYPE, NLICRESTRICT, NIDSTATE, NIDNUMBER, VEH\_ID, VLICENSETAG, VSTATE, VVEHYEAR, VCOLOUR, VMAKE, VMODEL, VVEHTYPE, VDECAL, VVIN, FINE, COURTDATE, JUDGE, COURT, DISPOSITION, COMMENT, CREATED, MODIFIED, CREATOR, ARRESTED, VIDEO FROM CITATIONS WHERE MODIFIED >= CAST('%%BEGIN\_DATE%%' AS DATE) AND MODIFIED <= CAST('%%END\_DATE%%' AS DATE);FILE=citations\_test.txt

[DSN=PROCOP]

UID=admin  
PWD=12345  
SRV=  
SQL=SELECT RECNO, ACCNUMBER, ACCTIME, BADGE, OFFICERNAME, ETIME, UTIME, NARRATIVE FROM ACCIDENTS WHERE (Format(acctime, 'yyyymmdd') > '%%BEGIN\_DATE%%' and Format(acctime, 'yyyymmdd') < '%%END\_DATE%%') or (Format(etime, 'yyyymmdd') > '%%BEGIN\_DATE%%' and Format(etime, 'yyyymmdd') < '%%END\_DATE%%') or (Format(utime, 'yyyymmdd') > '%%BEGIN\_DATE%%' and Format(utime, 'yyyymmdd') < '%%END\_DATE%%') order by recno desc;FILE=ACCIDENTS.DAT  
SQL=SELECT RECNO, ACCLINK, LNAME, FNAME, MNAME, AGE, DOB, SSN, HT, WT, RACE, SEX, HAIR, EYES, COMPLEX, BUILD, ADDRESS, CITY, STATE, ZIPCODE, HPHONE, WPHONE, WEXT, OCCUP, EMP, EMPADDR, EPHONE, NEXTKIN, KINADDR, KPHONE, DLN, DLSTATE, RESIDENT, ETIME, UTIME FROM ACCNAME WHERE (Format(etime, 'yyyymmdd') > '%%BEGIN\_DATE%%' and Format(etime, 'yyyymmdd') < '%%END\_DATE%%') or (Format(utime, 'yyyymmdd') > '%%BEGIN\_DATE%%' and Format(utime, 'yyyymmdd') < '%%END\_DATE%%') order by recno desc;FILE=ACCNAME.DAT

# Scheduler

**DBExtractor** can be run as a scheduled batch job. **DBExtractor** can be run as a scheduled task on any Windows operating system since there is a default Task Scheduler installed as part of the operating system. Task Schedulers show the exit status codes of programs they kick off.

While a scheduler provides some information on the reason for failed tasks, much more useful information can be found in the **DBExtractor** logging file. See the [logging](#) section for more details.

**DBExtractor** exit status codes:

**0 - SUCCESS**

**1 - FAILURE**