



**Faircom**<sup>®</sup>

c-tree **Plus**<sup>®</sup>  
**V9**

**c-tree ODBC  
Driver**

**c-tree Plus Edition Guide**



# **c-tree ODBC Driver**

c-tree Plus Edition Guide



Copyright © 1992-2008 FairCom Corporation All rights reserved. No part of this publication may be stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of FairCom Corporation. Printed in the United States of America.

Information in this document is subject to change without notice.

## Trademarks

c-tree, c-tree Plus, r-tree, the circular disk logo, and FairCom are registered trademarks of the FairCom Corporation. c-treeACE SQL, c-treeACE SQL ODBC, c-treeACE SQL ODBC SDK, c-treeVCL/CLX, c-tree ODBC Driver, c-tree Crystal Reports Driver, c-treeDBX, and c-treePHP are trademarks of FairCom Corporation. The following are third-party trademarks: AMD and AMD Opteron are trademarks of Advanced Micro Devices, Inc. Macintosh, Mac OS, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. Borland, the Borland Logo, Delphi, C#Builder, C++Builder, Kylix, and CLX are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. Business Objects, the Business Objects logo, Crystal Reports, and Crystal Enterprise are trademarks or registered trademarks of Business Objects SA or its affiliated companies in the United States and other countries. DBstore is a trademark of Dharma Systems, Inc. HP and HP-UX are registered trademarks of the Hewlett-Packard Company. AIX, IBM, OS/2, OS/2 WARP, and POWER5 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. Intel, Itanium, Pentium and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. LynuxWorks, LynxOS and BlueCat are registered trademarks of LynuxWorks, Inc. Microsoft, the .NET logo, MS-DOS, Visual Studio, Windows, Windows Mobile, Windows server and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Novell and NetWare are registered trademarks of Novell, Inc., in the United States and other countries. QNX and Neutrino are registered trademarks of QNX Software Systems Ltd. in certain jurisdictions. Red Hat and the Shadow Man logo are registered trademarks of Red Hat, Inc. in the United States and other countries, used with permission. SCO and SCO Open Server, are trademarks or registered trademarks of The SCO Group, Inc. in the U.S.A. and other countries. SGI and IRIX are registered trademarks of Silicon Graphics, Inc., in the United States and/or other countries worldwide. Sun, Sun Microsystems, the Sun Logo, Solaris, SunOS, JDBC, Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX and UnixWare are registered trademarks of The Open Group in the United States and other countries. Linux is a trademark of Linus Torvalds in the United States, other countries, or both. All other trademarks, trade names, company names, product names, and registered trademarks are the property of their respective holders.

FairCom welcomes your comments on this document and the software it describes. Send comments to:

Documentation Comments  
FairCom Corporation  
6300 W. Sugar Creek Drive  
Columbia, MO 65203

Portions © 1987-2008 Dharma Systems, Inc. All rights reserved. This software or web site utilizes or contains material that is © 1994-2007 DUNDAS DATA VISUALIZATION, INC. and its licensors, all rights reserved.

6/26/2008

# CONTENTS

---

<b>Quick Start</b> .....	<b>1</b>
1.1 ODBC History/Overview .....	1
1.2 Installation .....	1
1.3 Tutorial Setup.....	1
c-tree Server Tutorial Preparation .....	2
1.4 Tutorial .....	4
1.5 Tutorial File Layout .....	6
<b>Getting Started with Your Data</b> .....	<b>9</b>
2.1 Installing a Vendor Supplied Script.....	9
2.2 Browse Method .....	10
2.3 Script Method.....	11
2.4 Additional tips for the ctree Server Script .....	12
<b>ODBC/Application Setup</b> .....	<b>13</b>
3.1 ODBC Setup Dialog Options.....	14
Options>>.....	15
3.2 FairCom Data Dictionary - In Depth.....	17
Defining Multiple Databases .....	17
Adding Files to the Dictionary .....	18
3.3 Generic ODBC Compliant Application Notes.....	18
<b>c-tree ODBC Driver Technical Details</b> .....	<b>19</b>
4.1 c-tree ODBC Driver Purpose .....	19
4.2 Driver Types.....	19
4.3 Requirements.....	20
4.4 ODBC Compliance.....	21
4.5 Scalar Functions .....	21
4.6 SQL Conformance .....	23
Minimum SQL Grammar .....	23
Core SQL Grammar.....	24
Extended SQL Grammar .....	26
4.7 Driver Constraints .....	28
Limitations of SQL_LONGVARCHAR:.....	29
4.8 c-tree ODBC Driver Licensing.....	29
<b>Advanced Topics</b> .....	<b>31</b>
5.1 Performance Optimization .....	31
5.2 Data Dictionary Creation - Import Method .....	31
Import Script.....	32
Wildcard .....	32
Multiple Database Script.....	32
5.3 File Locations.....	32
32-bit Microsoft Supplied Files.....	33
32-bit FairCom Supplied Files.....	33
FairCom Supplied Sample Files .....	33
5.4 Advanced - Registry Options .....	34
Adding a Shareable DSN.....	35
Force Close Option .....	36
Japanese two-byte character (EUC) support.....	36
Optionally Disable Floating Point Unit Exceptions.....	36

5.5 DSN-less ODBC connections .....	36
<b>Errors .....</b>	<b>39</b>
6.1 DBOPEN DICTDBGETBYNAME(FAIRCOM.DB)=101 .....	39
6.2 No such database .....	39
6.3 No table names appear in the Select Table dialog box .....	39
6.4 Not able to open *.FCS file .....	39
6.5 Specified driver could not be loaded.....	39
6.6 Update failed. Continue editing record? .....	40
6.7 System Catalog's Files Need to be Rebuilt .....	40
6.8 Value Required .....	40
<b>Glossary .....</b>	<b>41</b>
<b>Index .....</b>	<b>43</b>

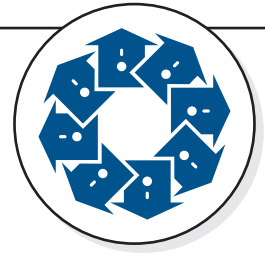
# FAIRCOM TYPOGRAPHICAL CONVENTIONS

Before you begin using this guide, be sure to review the relevant terms and typographical conventions used in the documentation.

The following formatted items identify special information.

Formatting convention	Type of Information
<b>Bold</b>	Used to emphasize a point or for variable expressions such as parameters.
CAPITALS	Names of keys on the keyboard. For example, SHIFT, CTRL, or ALT+F4.
<i>FairCom Terminology</i>	FairCom technology term.
<b>FunctionName()</b>	c-tree Function name.
<i>Parameter</i>	c-tree Function Parameter.
Code Examples	Code example or Command line usage.
<b>utility</b>	c-tree executable or utility.
<i>filename</i>	c-tree file or path name.
CONFIGURATION KEYWORDS	c-treeACE Configuration Keyword.
<b>BIG_ERR</b>	c-tree Error Code.





## Quick Start

Welcome to the c-tree® ODBC Driver - c-tree Plus® Edition. This powerful tool opens many possibilities for accessing and maintaining the data created by your c-tree Plus application. This guide is intended for the final user of the c-tree ODBC Driver and assumes an intermediate knowledge of the Microsoft Windows operating system.

This chapter provides a quick start with the c-tree ODBC Driver. It includes:

- A high-level overview of ODBC,
- Installation instructions, and
- An introductory tutorial.

### 1.1 ODBC History/Overview

Today's computer users demand direct and easy access to personal and corporate data using popular off-the-shelf applications. Microsoft Corporation's ODBC standard provides this type of open connectivity.

Database drivers that map a particular vendor's API (Application Programming Interface) to the ODBC standard API make this connection possible. This process is analogous to Windows printer drivers. The printer drivers allow an application developer to support virtually every printer. The developer programs to the Windows printer interface and the printer manufacturer provides a driver that works with the Windows interface to drive the printer. Though database portability is more complicated than the printer support the two are conceptually similar.

The c-tree ODBC Driver gives ODBC compliant applications, like Microsoft Access, Seagate Crystal Reports, and Visual Basic, access to c-tree Plus files.

### 1.2 Installation

Place the c-tree ODBC Driver disk into the appropriate drive. If Setup does not start automatically, execute **setup.exe** from the c-tree ODBC Driver disk and follow the instructions on the installation screens.

### 1.3 Tutorial Setup

Working through a quick tutorial increases familiarity with the operation of the Driver. This tutorial uses Microsoft Query. Even if you do not have Query, we still recommend reading this section and following along in your chosen ODBC application. You will find the information in this chapter helpful since most

ODBC applications share similar appearance and behavior with regard to defining and accessing ODBC Drivers.

A database can be thought of as a collection of related files. Large applications may have many different databases. Typically, files in the same database are related through c-tree Plus indices. The tutorial consists of a database of four data files and their associated indices. These files are described at the end of this chapter.

The c-tree ODBC Driver installs ready to use with the tutorial files. However, to run the tutorial with files controlled by the c-tree Server requires the preparation described below.

## c-tree Server Tutorial Preparation

Preparing the c-tree ODBC Driver for use with the c-tree Server involves the following simple steps:

1. Copying the sample data.
2. Creating the FairCom Data Dictionary.
3. Adjusting the c-tree ODBC Driver setup.

### Copy Sample Data

Copy the data and index files from the *FAIRCOM\ODBC\32bit\SAMPLE* directory (or the directory name supplied during installation) to the desired location on the c-tree Server machine. For the initial install, we recommend putting the files in the c-tree Server directory.

**Note:** If the c-tree Server is installed on a High/Low platform (i.e., Unix or Apple) use the sample data found in the directory *SAMPLEVHGHLOW*.

### Building a FairCom Data Dictionary Script

The ODBC interface requires a minimum set of information, a data dictionary, to access data files. The data dictionary includes a list of the data files in each database. The c-tree ODBC Driver automatically creates the FairCom Data Dictionary on the first access to the Driver by an ODBC compliant application. The Driver looks for a simple text script in the Data Dictionary Path specified in the c-tree ODBC setup. For each data file the script should contain a line consisting of the *alias\_name*, (the symbolic file name for the file), and the *file\_name*, (the actual file name as it resides on the c-tree Server machine hard drive). To simplify getting started, FairCom provides a sample script, *FAIRCOM.DB*, located in the c-tree ODBC Driver directory, *C:\FAIRCOM\ODBC\32bit* by default.

To adjust *FAIRCOM.DB* for use with the c-tree Server, use a text editor to prefix the sample file names with the fully qualified path from the c-tree Server perspective. For example: If you copied the files to the directory *C:\fairserv* in step 1, the script for the ODBC tutorial should look as follows:

```
CUSTOMERS      C:\fairserv\custmast.dat
ORDERS         C:\fairserv\custordr.dat
ORDER_DETAIL  C:\fairserv\ordritem.dat
ITEMS         C:\fairserv\itemmast.dat
```

The case sensitivity for the file and path names will be based on the requirements of the c-tree Server host machine operating system. If your c-tree Server is on a Unix-based operating system, use a forward slash, '/', as a path separator instead of a back slash, '\.

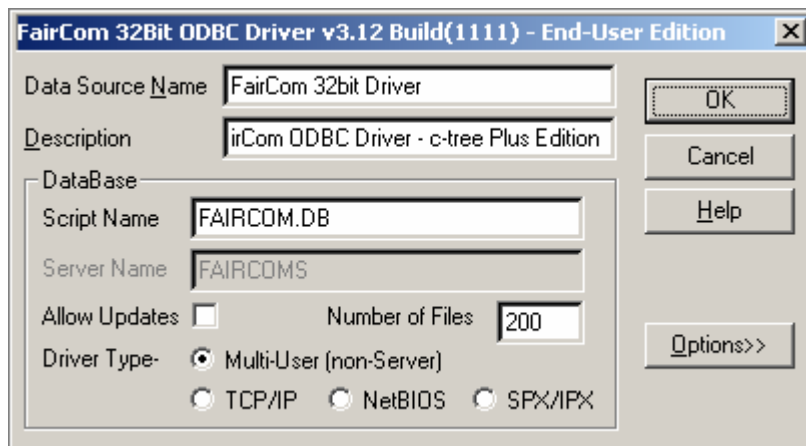
With a text script in place, such as *FAIRCOM.DB*, when the first ODBC compliant application accesses the c-tree ODBC Driver the data dictionary is created from the information in the text script.

## c-tree ODBC Driver Setup

To continue the tutorial, modify the c-tree ODBC Driver Setup options to match the c-tree Server to be used:

1. In the Windows Control Panel, open the ODBC manager.
2. Choose the FairCom ODBC Driver.
3. Click **Configure**.
4. The c-tree ODBC Setup dialog box shown as follows should appear.

**Figure 1: c-tree ODBC Setup dialog box**



The c-tree ODBC Setup dialog box contains the following prompts:

### Data Source Name

Arbitrary name of the c-tree Driver.

### Description

Arbitrary driver description.

### Script Name

The script name should be *FAIRCOM.DB* for the tutorial.

### Driver Type

The c-tree ODBC Driver makes it possible to select different operational modes for the Driver. The types are as follows:

Multi-User	FairCom's multi-user non-server ( <i>FPUTFGET</i> ) mode of operation.
TCP/IP	Communicate through the TCP/IP communication protocol to a c-tree Server.
NETBIOS	Communicate through the NETBIOS communication protocol to a c-tree Server.
SPX/IPX	Communicate through the SPX/IPX communication protocol to a c-tree Server.

**Note:** Select the proper communication protocol. If you are unsure which protocol to select, start with TCP/IP, which is the default for the c-tree Server, or contact your Server Administrator.

## Server Name

Default: FAIRCOMS

The default c-tree Server name may be overridden with the `SERVER_NAME` keyword in `ctsvr.cfg`, the Server configuration file located in the Server directory. If you are not sure of the name of your c-tree Server, contact your Server Administrator.

When using TCP/IP, the Server Name must include the host machine name. Both the Server Name and the host name are case sensitive. The format is as follows, where `Server_Name` is the name of the c-tree Server, and `Server_Hostname` is the name of the machine where the c-tree Server resides. If you are unsure about the host name, consult your network administrator:

`Server_Name@Server_Hostname`

For example, to connect to a machine with a host name of Sun and using the default Server Name, FAIRCOMS, specify the following in the ODBC Setup Server Name prompt: FAIRCOMS@Sun

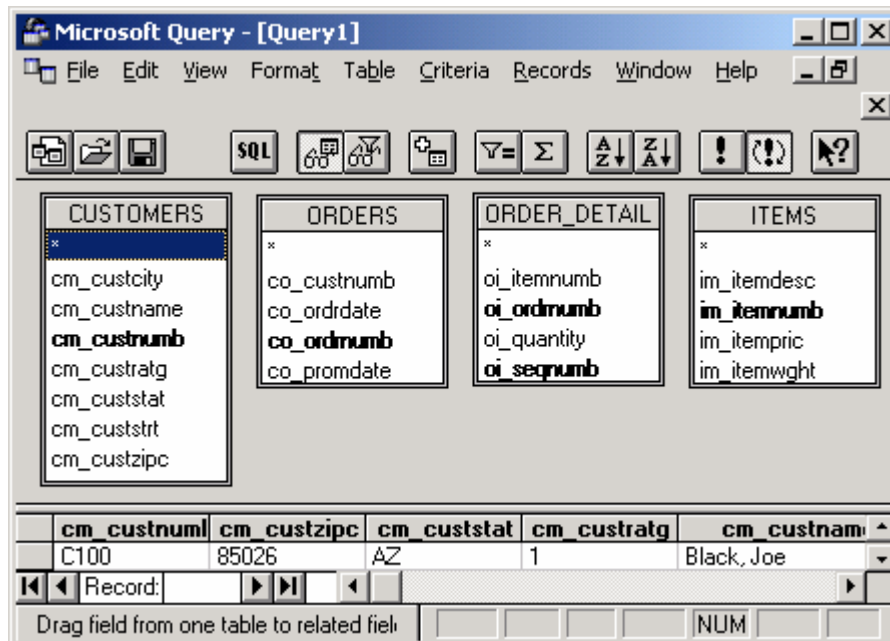
Once the Server Name and other prompts are properly defined, click **OK** for the Setup and ODBC Data Source Administrator dialog boxes. To continue the tutorial, please proceed to ["Tutorial"](#).

## 1.4 Tutorial

1. Start Microsoft Query (double click the Microsoft Query icon).
2. Select **New...** from the Microsoft Query File menu.
3. From the Choose Data Source dialog box, choose the Databases tab. For simplicity, remove the checkmark from "Use the Query Wizard to create/edit queries". Highlight the c-tree ODBC Driver, and click OK.
4. In the Add Tables dialog box listing the four sample files, continue by adding all four files in the order shown in [Figure 1-2: Microsoft Query Active Window on page 1-5](#) (Choose each file and select Add). After adding all four files click Close. The Microsoft Query active window should appear similar to [Figure 1-2: Microsoft Query Active Window on page 1-5](#).

Figure 1-2: Microsoft Query Active Window

Equation 1: -2: Microsoft Query Active Window

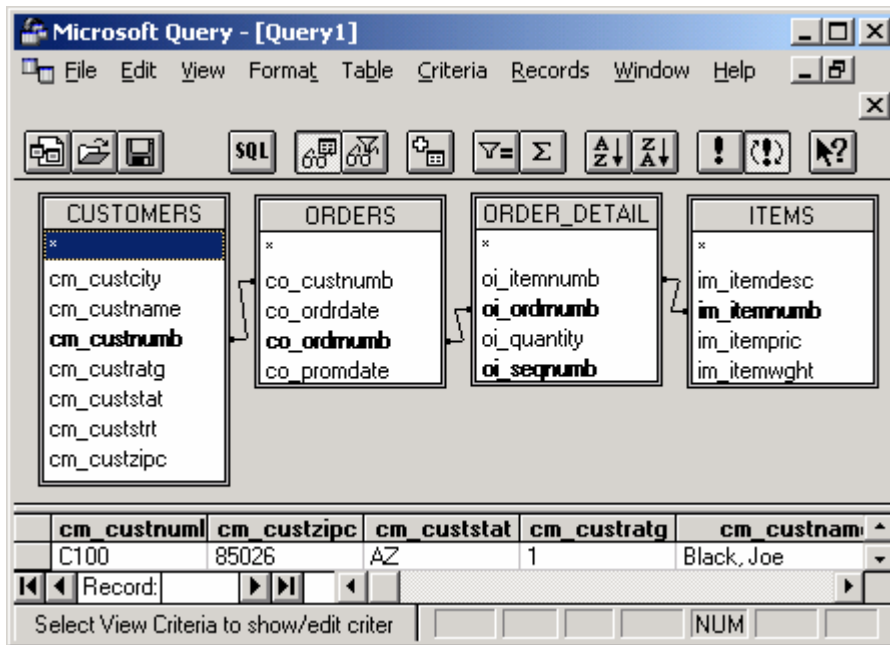


5. Add file relationships between the four by selecting the cm\_custnumb field in the CUSTOMERS file by clicking and holding the left mouse button. Drag the cursor over to the co\_custnumb field in the ORDERS file and release the left mouse button. This should draw a line between the two fields, as shown in [Figure 1-3: Microsoft Query Active Window with Lines on page 1-6](#). Using [Figure 1-3: Microsoft Query Active Window with Lines on page 1-6](#) and the following table, relate the ORDERS file to ORDER\_DETAIL and ORDER\_DETAIL to the ITEMS file.

Data File1	Common Field	Data File 2	Common Field
CUSTOMERS	cm_custnumb	ORDERS	co_custnumb
ORDERS	co_ordnumb	ORDER_DETAIL	oi_ordnumb
ORDER_DETAIL	oi_itemnumb	ITEMS	im_itemnumb

Figure 1-3: Microsoft Query Active Window with Lines

**Equation 2: -3: Microsoft Query Active Window with Lines**



6. Once the files are related, double-click any field to retrieve the data for that file. Since the files are all related, the corresponding data across files will be properly aligned.

**Note:** To select all fields for a particular file using Microsoft Query, double click the asterisk (\*) at the top of the file box.

7. The last step for this tutorial is to insert (write) a few new records into a data file. By default, all c-tree ODBC Drivers are read ONLY. It is possible to enable writes for all but the c-tree Read ONLY Driver, but by doing so you are taking responsibility for the integrity of your data. The authors of your c-tree Plus application have taken extreme care to ensure the integrity of your data files. For example, in an accounting system, an order total may be the sum of several individual products. If you override the price of one of the products, and do not adjust the order total accordingly, the data integrity could be compromised. If you are prepared to take responsibility for the integrity of your data, the c-tree ODBC Driver can be enabled for writing by using the ODBC Setup icon (see [“ODBC/Application Setup”](#)). If you are unsure about the ramifications of enabling the write feature, consult the software vendor of your c-tree Plus application. If you decide to enable the write capability, the following steps may be used to verify its operation. Microsoft Query only allows single file updates. For this reason, do the following:

- a) Remove all but one of the data files from your query. We deleted Orders, Order\_Detail, and Items. This is done by highlighting the file to be removed and pressing the delete key.
- b) Pull down the Records menu and enable Allow Editing.
- c) Display the desired fields.
- d) Using the Customers scroll bar, move to the end of the Customers data. The last record in a file enabled for inserts will have an asterisk (\*) in the first column.
- e) Enter data into the desired fields and then press Enter to insert the data. Once the last field has been filled in, press the Enter or Tab key to go to the first column of the next row. This inserts the new data and updates the record count in the bottom left corner of the dialog box.

## 1.5 Tutorial File Layout

The Tutorial uses the following four data files and six indices:

Data File	Symbolic Index Name	Key Segment(s)	Unique
custmast.dat	cm_custnumb_idx	cm_custnumb	yes
itemmast.dat	im_itemnumb_idx	im_itemnumb	yes
custordr.dat	co_ordrnumb_idx	co_ordrnumb	yes
custordr.dat	co_custnumb_idx	co_custnumb	no
ordritem.dat	oi_ordrnumb_idx	oi_ordrnumb, oi_seqnumb	yes
ordritem.dat	oi_itemnumb_idx	oi_itemnumb	no

**Note:** The unique column indicates if the index accepts duplicate data values.

*custmast.dat*, the Customer Master File, contains the table CUSTOMERS made up of the fields shown below:

Symbolic Name	Field Description	Field Type	Field Length
cm_custnumb	Customer number	CT_STRING	5
cm_custzipc	Customer zip code	CT_STRING	10
cm_custstat	Customer state	CT_STRING	3
cm_custratg	Customer rating	CT_STRING	2
cm_custname	Customer name	CT_STRING	48
cm_custadr	Customer address	CT_STRING	48
cm_custcity	Customer city	CT_STRING	48

*itemmast.dat*, the Item Master File, contains the table ITEMS made up of the fields shown below:

Symbolic Name	Field Description	Field Type	Field Length
im_itemwght	Item weight	CT_INT4	4
im_itempric	Item price	CT_MONEY	4

Symbolic Name	Field Description	Field Type	Field Length
im_itemnumb	Item number	CT_STRING	6
im_itemdesc	Item description	CT_STRING	48

*custordr.dat*, the Customer Order File, contains the table ORDERS made up of the fields shown below:

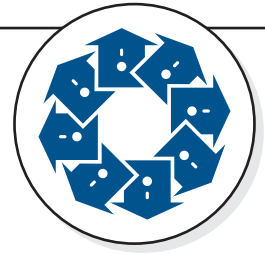
Symbolic Name	Field Description	Field Type	Field Length
_o_delflag	Order delete flag	CT_INT4	4
co_ordrdate	Order date	CT_DATE	4
co_promdate	Order promise date	CT_DATE	4
co_ordrnumb	Order number	CT_STRING	7
co_custnumb	Customer number	CT_STRING	5

*ordritem.dat*, the Order Item File, contains the table ORDER\_DETAIL made up of the fields shown below:

Symbolic Name	Field Description	Field Type	Field Length
_i_delflag	Order item delete flag	CT_INT2	2
oi_seqnumb	Order item stock number	CT_INT2	2
oi_quantity	Order item quantity	CT_INT2	2
oi_ordrnumb	Order item number	CT_STRING	7
oi_itemnumb	Item number	CT_STRING	6

**Note:** The fields beginning with “\_”, *\_o\_delflag* and *\_i\_delflag*, are hidden fields and will not be displayed. Developers looking for more information on other field options should refer to Section 1.5 Field Options in *c-tree Driver Developer’s Guide*.





## Getting Started with Your Data

The FairCom Data Dictionary, the Driver's catalog of available data files, created in the previous chapter was specific to the sample tutorial. This chapter helps you get an ODBC compliant application operational with your c-tree Plus data. To take full advantage of the c-tree ODBC Driver features, consult "[ODBC/Application Setup](#)" and "[Advanced Topics](#)" for additional application setup information and performance tips.

- There are a few ways to provide the information needed to build the data dictionary: with a vendor-supplied script, by browsing for the files, or with a locally-created script.
- If your vendor provided a script, this is the simplest solution. Proceed to "[Installing a Vendor Supplied Script](#)".
- To simply browse your disks for the necessary files, proceed to "[Browse Method](#)".
- If you know the names and locations of the files and would like to create a script, proceed to "[Script Method](#)".

### 2.1 Installing a Vendor Supplied Script

All ODBC drivers need some form of data dictionary to describe which files are in each database. To make the c-tree ODBC Driver easier to install, FairCom allows your software vendor to create the FairCom Data Dictionary script for you.

- If you purchased the c-tree ODBC Driver from your c-tree Plus application vendor, check the c-tree ODBC Driver package for an existing script file named *VENDOR.DB*. If *VENDOR.DB* is present, skip to the next paragraph.
- If *VENDOR.DB* is not present, or you did not purchase the c-tree ODBC Driver from your c-tree Plus application vendor, skip to "[Browse Method](#)". to browse or "[Script Method](#)". to create your own script.

If your vendor provides *VENDOR.DB*, check the pocket inside the cover of this guide for a document titled, "Notes from your Software Provider". If this document is present, it contains additional instructions from your software vendor. To create the dictionary, copy *VENDOR.DB* to the data dictionary path, *C:\FAIRCOM\ODBC\32bit* by default, and rename it to the Script Name, *FAIRCOM.DB* by default, as follows:

```
COPY A:\VENDOR.DB C:\FAIRCOM\ODBC\32bit\FAIRCOM.DB
```

**Note:** You may receive a warning message stating that *FAIRCOM.DB* already exists. Most likely, an existing *FAIRCOM.DB* is left over from the Tutorial, in which case you can answer the prompt with a Yes to replace the file.

The FairCom Data Dictionary (*CTSYSCAT.FCS*) will now automatically be created upon the first access to the c-tree ODBC Driver by an ODBC compliant application. Once the dictionary is created, you may delete *FAIRCOM.DB*.

## 2.2 Browse Method

This method allows you to browse in search of the c-tree Plus files to be included in the FairCom Data Dictionary. Enable this feature by specifying a wild card identifier in the Script Name setting of the c-tree ODBC Driver setup window, as follows:

1. Access the c-tree ODBC Driver setup window by selecting the ODBC icon from the Control Panel. Highlight the appropriate c-tree ODBC Driver and click **Configure....**
2. In the c-tree ODBC Driver setup window, enter a desired database file name filter by inserting a file name and wildcard combination (as discussed below) into the Script Name prompt. The two standard wild cards supported are as follows:

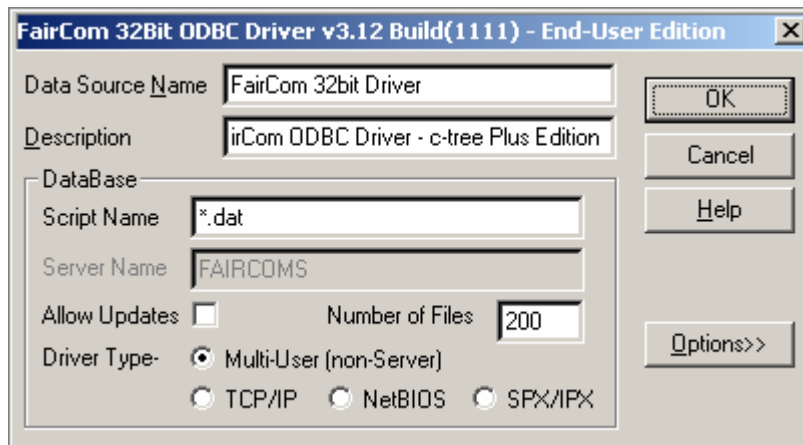
- \* match all characters
- ? match a single character

For example, with the following c-tree Plus files: *ABBC.DAT*, *ABCD.DAT*, and *ABDD.DAT*.

- Entering AB\* matches all three files.
- Entering AB?D.DAT matches *ABCD.DAT* and *ABDD.DAT* only.

Figure below shows a wildcard search for \*.dat, finding all files that have a .dat file extension.

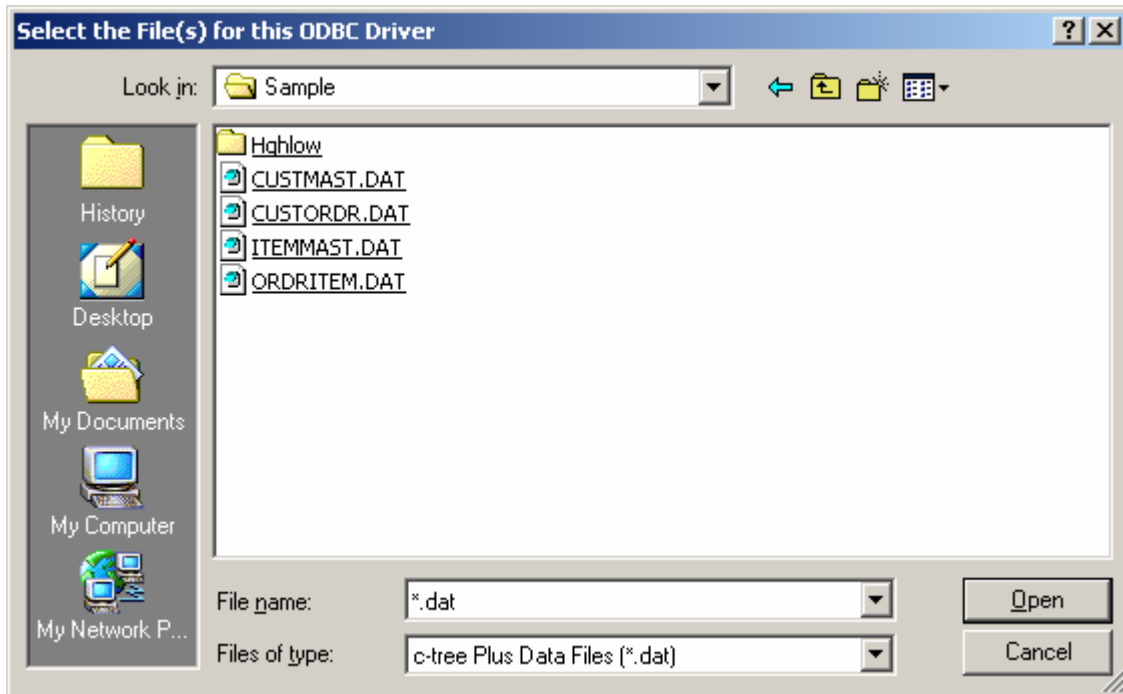
**Figure 2: Wildcard Search**



3. Once the Script Name setting contains a wildcard, close the Setup window by clicking **OK**. In the Data Sources window, select **OK**.

- The first ODBC compliant application that selects the c-tree ODBC Driver will cause a Database File Selection window to appear asking if you “want to browse for the files to be used by this ODBC Driver”. By answering Yes, you are presented with the screen listed below in the figure below.

**Figure 3: Selecting Files**



- From the File Browse window, highlight the files you would like available to the c-tree ODBC Driver. Please note the following two points:
  - To select multiple files, hold down the shift key or control key while selecting files.
  - Files can be selected only from one directory at a time. If you have files in multiple directories, select all of the files from the first directory then select OK. After inserting these files, the c-tree ODBC Driver will ask if you wish to select additional files. If so, this will be your opportunity to include files from other directories and append them into one common FairCom Data Dictionary.
- Once you have selected all of the files, answer “No” to indicate that there are no more files to select. The c-tree ODBC Driver will automatically create the FairCom Data Dictionary, *CTSYSCAT.FCS*.

## 2.3 Script Method

By creating a simple text script, the FairCom Data Dictionary can be created automatically upon the first access to the c-tree ODBC Driver by an ODBC compliant application. A sample text script, *FAIRCOM.DB*, is included and is placed in the installation directory (*C:\FAIRCOM\ODBC\32bit* by default). The format of the text script is as follows. *alias\_name* is the symbolic file name referenced from an ODBC application. *file\_name* is the actual file name as it resides on disk, including the directory structure.

```
alias_name file_name
```

The text file should list the file alias names and the actual file names on a separate line for each file (see [“Additional tips for the ctree Server Script”](#) for an example).

The steps to create the FairCom Data Dictionary using the script method are:

- Create the script with the format defined above.
- Specify the name of the script with the Script Name setting in the c-tree ODBC Driver setup window.

3. Set the Data Dictionary Path setting in the Options section of the c-tree ODBC Driver setup window to the directory where the script resides.
4. With the script in place, when the first ODBC compliant application accesses the c-tree ODBC Driver, the FairCom Data Dictionary, *CTSYSCAT.FCS*, will be created from the information located in the script file. The FairCom Data Dictionary will be created in the Data Dictionary Path in multi-user non-server mode or in the c-tree Server directory when using the c-tree Server.

After successfully creating the dictionary, the text script is no longer needed.

**Note:** A third method for creating the FairCom Data Dictionary is available for advanced users. This method is to use the `import.exe` utility shipped with the c-tree ODBC Driver. See [“Data Dictionary Creation - Import Method”](#) for additional information.

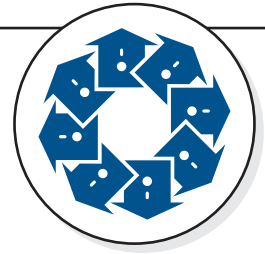
## 2.4 Additional tips for the ctree Server Script

The following suggestions are intended for c-tree ODBC Driver dictionary scripts when used with the c-tree Server.

It is best to either prefix the file name with the path to the data files, from the c-tree Server perspective, or include no path at all. For example, to create the data dictionary for a c-tree Server residing in the directory `/usr/fairserv`, `FAIRCOM.DB` should appear as here:

```
CUSTOMERS      /usr/fairserv/custmast.dat
ORDERS         /usr/fairserv/custordr.dat
ORDER_DETAIL   /usr/fairserv/ordritem.dat
ITEMS          /usr/fairserv/itemmast.dat
```

The case sensitivity for file and path names is based on the requirements of the operating system (i.e., most Unix file naming conventions are case sensitive). If your c-tree Server is on a Unix based operating system (AIX, SCO, AT&T, etc.) be sure to use a forward slash '/' to separate the paths as shown in Figure 2-3: faircom.db - Data Dictionary for Server in `/usr/fairserv` on page 2-6.



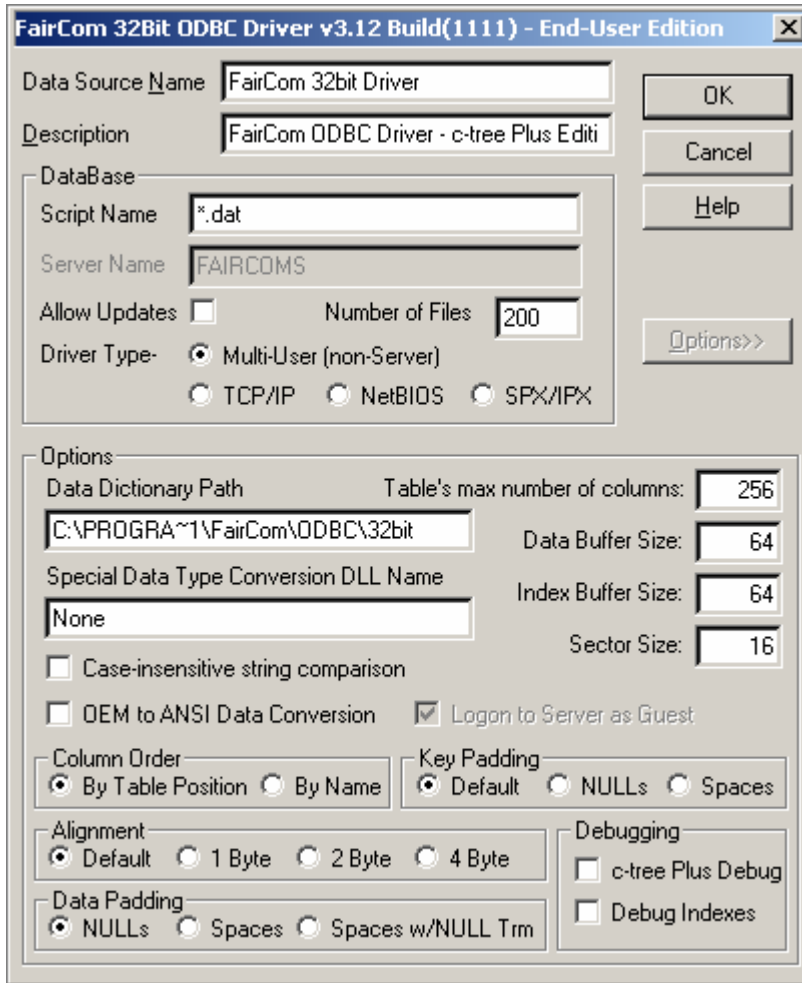
## ODBC/Application Setup

This chapter details the c-tree ODBC Driver Setup dialog and provides valuable insight into optimal application setup. This information is important for understanding the features and options required to effectively use the c-tree ODBC Driver. Contact your application vendor for information on changing the default settings.

### 3.1 ODBC Setup Dialog Options

The c-tree ODBC Driver is optimized to simplify setup. The c-tree ODBC Setup dialog box is accessible from the Windows Control Panel ODBC icon or your ODBC compliant application and provides the following options as shown in the following:

**Figure 4: Setup Dialog Box**



**Data Source Name**

Arbitrary database (application view) name.

**Description**

Arbitrary description of database.

**Script Name**

Default: *FAIRC.COM.DB*

Database script file name. See [“Building a FairCom Data Dictionary Script”](#). This prompt is not case sensitive. See additional notes in [“FairCom Data Dictionary - In Depth”](#).

## Server Name

Default: FAIRCOMS

The c-tree Server name. The default may be overridden with the `SERVER_NAME` keyword in `ctsrvr.cfg`. If you are not sure of the name of your c-tree Server, contact your network administrator or consult `ctsrvr.cfg` in your c-tree Server directory.

When using TCP/IP, the Server Name must include the c-tree Server host machine name. Both the Server Name and host name are case sensitive. The format is: `Server_Name@Server_Hostname`, where `Server_Name` is the name of the c-tree Server and `Server_Hostname` is the name of the machine running the c-tree Server. Consult your network administrator for more information.

For example, to connect to a machine with a hostname of Sun and using the default Server name FAIRCOMS, specify the following Server Name:

FAIRCOMS@Sun

## Allow Updates

Default: NO

Enabling this feature allows writes, or inserts, into files available to the c-tree ODBC Driver.

**Note:** By enabling this option, you will be responsible for maintaining the integrity of your c-tree Plus data. The authors of your c-tree Plus based application have taken extreme care to ensure the integrity of your c-tree Plus data files. For example, in an accounting system, an order total may be the sum of several individual products. If you override the price of one of the products and do not adjust the order total accordingly, the data integrity could be compromised. If you are unsure about the ramifications of enabling this write feature, consult the software vendor from whom your c-tree Plus based application was purchased.

## Number of Files

Default: 200

Maximum number of concurrently open c-tree Plus files (data plus index).

## Driver Type

Select the I/O operational mode for the c-tree ODBC Driver. The choices are:

Driver Type	Explanation
Multi-User	FairCom's multi-user non-server ( <i>FPUTFGET</i> ) mode
TCP/IP	Communicate through the TCP/IP communication protocol to a c-tree Server.
NETBIOS	Communicate through the NETBIOS communication protocol to a c-tree Server.
SPX/IPX	Communicate through the SPX/IPX communication protocol to a c-tree Server.

## Options>>

The **Options>>** button is intended for developers or advanced users only. By selecting the **Options>>** button, it is possible to specify several initialization parameters. For the c-tree tutorial and the first execution of your ODBC compliant application, FairCom recommends using the default values. The following options are available:

### Data Dictionary Path

Default: installed path

FairCom Data Dictionary location in multi-user mode. When using the c-tree Server, the dictionary is created in the c-tree Server directory and should remain there.

The c-tree ODBC Driver searches the Data Dictionary Path for the data dictionary script specified in Script Name. If found, the contents of the script are used to recreate *CTSYSCAT.FCS*.

### Special Data Type Conversion DLL Name

This entry has been added to allow ODBC developers to add support for their own data types with the c-tree Driver SDK.

### Max columns per table

Default: 256

Specifies the maximum number of columns to be supported by a given file. The maximum is presently 5000. However, please note that some ODBC compliant applications limit this value to 256 or 300. Even though the c-tree ODBC Driver can open tables with up to 5000 columns, only 300 columns may be specified in a given SQL query select list.

### Buffer Sizes - Data/Index

Default: 64

These values set the amount of memory for data and index caching in multi-user non-server mode only. Typically, the larger the value the better the performance. The memory is calculated as follows: Bytes in RAM = (Sector Size X Buffer Size X 128).

### Sector Size

Default: 16

Index node size. Set the same value as the target c-tree Plus files, or larger if they are not superfiles. The default value of 16 yields a 2K node size (16 x 128), the default for the c-tree Server. Each time this value is changed, *CTSYSCAT.FCS* must be recreated to match the new value. Normally, the default value will be best, but your application vendor will know the optimum value for your c-tree Plus files.

### Case-insensitive string comparison

Default: OFF

When enabled, causes target string values to be compared to column data without regard to case. For example, "THE" and "the" are identical with this box checked.

### OEM to ANSI Data Conversion

Default: OFF

Allows customers to choose to perform the OEM to ANSI conversion for string data. Consult your vendor before changing this setting.

### Logon to Server as Guest

Default: ON

Username and password support for client-server mode. When enabled in client-server mode, the c-tree ODBC Driver connects to the c-tree Server as user GUEST. If this option is not enabled, the user is prompted for the username and password when connecting to the c-tree Server.

### Column Order

Default: Table Position

This prompt specifies how column names (field names) are displayed. The default method lists the columns in the order they appear in the table. Alternatively, the column names can be ordered By Name, which sorts them in alphabetical order before displaying.

### Key Padding

Default: Default

This option allows the padding byte for target key values to be altered. Note: This option should only be used by advanced users.

### Alignment

Default: Default

This option allows the default c-tree Plus data file alignment to be altered. Note: This option should only be used by advanced users.

### Data Padding

Default: NULLs

This option allows the padding byte for string data to be altered. Note: This option should only be used by advanced users.

### c-tree Plus Debug

Default: OFF

Enabling this feature sends debugging information to a log file named *CTODBC.LOG*.

### Debug Indexes

Default: OFF

Enabling this feature sends debugging information to a log file named *CTODBC.LOG*.

## 3.2 FairCom Data Dictionary - In Depth

This section contains additional information about the FairCom Data Dictionary. The browse and script methods for creating the FairCom Data Dictionary are found in ["Building a FairCom Data Dictionary Script"](#). Instructions include defining multiple databases, adding files to the dictionary, and general ODBC notes.

### Defining Multiple Databases

One important concept is that each database defined by the Script Name setting in the c-tree ODBC Setup dialog can be thought of as a view into a larger database. The FairCom Data Dictionary (*CTSYSCAT.FCS*) can hold many database names (or views), each containing many file (table) names. The fewer files in each database, the faster data access will be. It is advisable to split your application files across many databases, each containing different groups of files.

For example, the tutorial has four files: Customer, Orders, Order\_Detail, and Items. If a particular user only requires access to information from the Customer file, create a database with only the Customer

file. This is done with the automatic dictionary method by creating a dictionary script with just the customer file, *CUSTOMER.DB*, as follows:

```
CUSTOMERS      c:\FAIRCOM\ODBC\32bit\sample\custmast.dat
```

Using the ODBC Setup icon in the Windows Control Panel, a new ODBC Driver entry can be created that specifies the Script Name to be *CUSTOMER.DB*.

## Adding Files to the Dictionary

To add additional files to the FairCom Data Dictionary, create a new dictionary script listing all files to include in that database. The new files will be appended to the database name that corresponds to the script name. This also allows you to remove files from a database. The re-creation of the dictionary takes place the next time the c-tree ODBC Driver is accessed.

For example, to add a new file *ORDERS* to *CUSTOMER.DB*, a script would be created as follows:

```
CUSTOMERS      c:\FAIRCOM\ODBC\32bit\sample\custmast.dat  
ORDERS         c:\FAIRCOM\ODBC\32bit\sample\order.dat
```

For this example, ensure the Script Name is set to *CUSTOMER.DB* and the location specified by the Data Dictionary Path is the location of *CUSTOMER.DB*. The next time an ODBC application accesses the c-tree ODBC Driver that references *CUSTOMER.DB* as the Script Name, the *ORDERS* file will be appended to the dictionary. After the file is appended to the dictionary, the script (*CUSTOMER.DB*) is no longer needed.

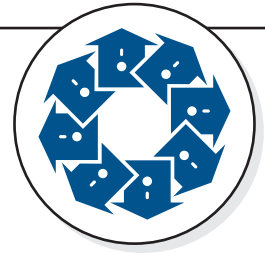
## 3.3 Generic ODBC Compliant Application Notes

ODBC compliant applications, such as Microsoft Query, keep their own internal list of supported drivers. Therefore, if you remove a driver from the Windows ODBC setup icon, the driver will still appear in the Microsoft Query internal driver list. See your ODBC compliant application instructions for the specific procedures used to remove ODBC drivers from its internal list.

Most ODBC compliant applications provide a mechanism for searching to the first or last record of a file. If a multi-file search does not have file relationships defined, or if they are defined incorrectly, the search may take a long time. For example, if a report listing one field from each of the four files in the c-tree ODBC tutorial was executed without file relationships, it could take several minutes. The ODBC application will search all permutations of field cross-referencing in an effort to establish a proper link. To prevent extended search times, ALWAYS define file relationships for multi-file operations.

When creating multiple database references in the ODBC Dictionary, *CTSYSCAT.FCS*, it may be necessary to exit the ODBC compliant application in between creating dictionaries. For example, to create a dictionary entry named *CUSTOMER*, create a script named *CUSTOMER.DB* and place it in the directory specified by the Data Dictionary Path setting. Start an ODBC compliant application, such as Microsoft Query. When the ODBC Driver referencing *CUSTOMER.DB* as the Script Name is selected, the dictionary will have *CUSTOMER* entered as a valid database name. Before creating a new dictionary entry by referencing a new script, first exit the ODBC application (i.e., Microsoft Query in this example).

The c-tree 32-bit ODBC Driver configuration automatically creates a file DSN during installation, and updates it when saving the current ODBC driver configuration.



## c-tree ODBC Driver Technical Details

This chapter provides additional background and details for the c-tree ODBC Driver and is not considered required reading.

### 4.1 c-tree ODBC Driver Purpose

The c-tree ODBC Driver is a single-tier driver that interfaces directly to the c-tree Plus API. The term single-tier indicates the driver contains all of the program logic, including an SQL interpreter, to handle requests from a front-end application.

**Note:** References to c-tree Plus include the c-tree Server.

The c-tree ODBC Driver handles the application conversion processes necessary for ODBC compliant applications to access c-tree Plus files, including:

- Connecting the front-end application to a c-tree Plus database (data and index file(s)).
- Defining data - includes retrieving data file-specific information (such as file mode), and creating and deleting data and index files.
- Manipulating data - includes adding, deleting, retrieving and updating database files.
- Disconnecting from a c-tree Plus database.
- Performing general utility functions - includes retrieving extended error messages, comparing data for equality, etc.
- Processing transactions - includes logging, committing, and rollback.
- Process optimization - includes processing joins and filters.

### 4.2 Driver Types

The c-tree ODBC Driver is available in 16-bit and 32-bit implementations. The 16-bit implementation is designed for Windows 3.1, Windows for Workgroups 3.1x, Windows 95/98, and the Windows NT/2000 operating systems while the 32-bit Driver is designed for Windows 95/98 and Windows NT/2000 only. Both the 16-bit and 32-bit Drivers are available in the following configurations:

#### End-User Driver

Supports multi-user non-server and c-tree Server based applications. This Driver is configured by default for read-only access but can be enabled to support database writes for all modes of operation.

### **End-User Read-Only Driver**

Supports multi-user non-server and c-tree Server based applications, in read-only mode. This Driver does NOT provide the ability to perform database writes or to create tables. It is designed for read-only access to existing c-tree Plus files.

### **Developer-Only Driver**

c-tree Plus includes complementary 32-bit Development ODBC Drivers. These Drivers allow c-tree Plus application developers to test ODBC compliant programs with c-tree Plus data files. These Drivers are not licensed for production use or distribution and are identified by the "Developer Edition" dialog box displayed upon startup.

## **4.3 Requirements**

The c-tree ODBC Driver operates on any IBM PC or compatible machine (286 or higher) with at least 4 MB of RAM. 2 MB of available hard drive space is required for the 16-bit version or 3 MB for the 32-bit version.

The Driver is implemented as a dynamic link library, DLL, supporting standard and extended c-tree Plus file in stand-alone multi-user non-server mode and the FairCom c-tree Server version 6.X or later (V7 or later for Extended files). Your application vendor must also provided meta-data for the data files in the form of resources stored in the files or an external parameter file.

## 4.4 ODBC Compliance

The ODBC standard has three levels of compliance: Core, Level 1 and Level 2. The c-tree ODBC Driver fully supports Core and Level 1, and supports some ODBC Level 2 functions:

Core Functions	Level1 Functions	Level 2 Functions
SQLAllocConnect	SQLBindParameter	SQLBrowseConnect
SQLAllocEnv	SQLColumns	SQLDataSources
SQLAllocStmt	SQLDriverConnect	SQLDescribeParam
SQLBindCol	SQLGetConnectOption	SQLDrivers
SQLCancel	SQLGetData	SQLMoreResults
SQLColAttributes	SQLGetFunctions	SQLNativeSql
SQLConnect	SQLGetInfo	SQLNumParams
SQLDescribeCol	SQLGetStmtOption	SQLPrimaryKey
SQLDisconnect	SQLGetTypeInfo	
SQLError	SQLParamData	
SQLExecDirect	SQLPutData	
SQLExecute	SQLSetConnectOption	
SQLFetch	SQLSetStmtOption	
SQLFreeConnect	SQLSpecialColumns	
SQLFreeEnv	SQLStatistics	
SQLFreeStmt	SQLTables	
SQLGetCursorName		
SQLNumResultCols		
SQLPrepare		
SQLRowCount		
SQLSetCursorName		
SQLTransact		

The level of functionality supported by FairCom is consistent with other ODBC drivers in the marketplace and meets or exceeds the needs of most popular ODBC compliant applications.

## 4.5 Scalar Functions

The c-tree Plus ODBC Driver supports ODBC scalar functions which may be included in an SQL statement as a primary expression. For a description of the SQL syntax for these functions, see the ODBC Programmer's Reference.

### String Functions

CONCAT  
LCASE  
LEFT  
LENGTH

LOCATE  
LTRIM  
RIGHT  
RTRIM  
SPACE  
SUBSTRING  
UCASE

**Example**

```
SELECT emp_id FROM emp WHERE LCASE(emp_name) = 'karen jung'
```

**Numeric Functions**

ABS  
ACOS  
ASIN  
ATAN  
ATAN2  
CEILING  
COS  
COT  
DEGREES  
EXP  
FLOOR  
HOUR  
LOG  
LOG10  
MOD  
PI  
POWER  
RADIANS  
RAND  
ROUND  
SIGN  
SIN  
SQRT  
TAN  
TRUNCATE

**Example**

```
SELECT * FROM tbl1 WHERE MOD(col1, 100) = 0
```

**Date Functions**

CURDATE  
CURTIME  
DAYNAME  
DAYOFMONTH  
DAYOFWEEK  
DAYOFYEAR  
MINUTE  
MONTH  
MONTHNAME  
NOW  
QUARTER  
SECOND  
WEEK  
YEAR

**Example**

```
SELECT * FROM emp WHERE YEAR(hire_date) <1990
```

**System Functions**

DATABASE

IF

NULL

USER

**Example**

```
SELECT * FROM emp WHERE ename = USER()
```

The system scalar functions IF and NULL are SQL extensions.

IF allows you to enter different values depending on whether the condition is true or false. The syntax is:

```
IF(predicate, expression, expression)
```

For example, if you want to display a column with logical values as "True" or "False" instead of a binary representation, you would use the following SQL statement:

```
SELECT IF(logicalcol=1, "True", "False")
```

The system scalar function NULL allows you to set a column as null values. The syntax is:

```
NULL()
```

For example, the following SQL statement retrieves null values:

```
SELECT NULL() FROM emp
```

**Conversion Function**

CONVERT

**Example**

```
SELECT empno FROM emp WHERE hire_date > CONVERT('1993-01-02', SQL_DATE) + 30
```

## 4.6 SQL Conformance

The c-tree ODBC Driver fully supports the minimum SQL grammar and a portion of the core and extended SQL grammar requirements of the ODBC standard, as shown in the following tables:

### Minimum SQL Grammar

The minimum Grammar requirements (fully met by the c-tree ODBC Driver) are as follows:

- Create Table
- Delete (searched)
- Drop Table
- Insert
- Select
- Update (searched)

The following tables provides some examples of the minimum SQL grammar supported by the FairCom ODBC Driver:

Grammar	Examples	Comments
CREATE TABLE	CREATE TABLE sal (emp_id integer, name char(50), salary float, hire_date date)  CREATE TABLE emp (emp_id integer NOT NULL, PRIMARY KEY (emp_id) )	Column constraint definitions supported: NOT NULL.  Table constraint definitions supported: UNIQUE and PRIMARY KEY DEFAULT. Default-value is not supported.
DELETE	DELETE FROM sal WHERE name = 'John Smith'	
DROP TABLE	DROP TABLE sal	[CASCADE   RESTRICT] is not supported.
INSERT	INSERT INTO sal VALUES (34086, 'Fred Black', 45000.00, '1992-05-25')	
SELECT	SELECT * FROM sal  SELECT emp.emp_id, sal.salary FROM emp, sal WHERE emp.emp_id = sal.emp_id	
UPDATE	UPDATE sal SET salary = 35000.00 WHERE emp_id = 25089	

## Core SQL Grammar

The Core Grammar supported by the FairCom ODBC Driver is as follows:

- Create Index
- Create View
- Drop Index
- Drop View
- Select
  - Approximate numeric literal
  - Between predicate
  - Correlation name
  - Exact numeric literal
  - IN predicate
  - Set function
- Subqueries

The following tables provides some examples of the core SQL grammar supported by the FairCom ODBC Driver:

Grammar	Examples	Comments
CREATE INDEX	CREATE INDEX empidx ON emp (emp_id, emp_name)  CREATE UNIQUE INDEX empidx ON emp (emp_id, emp_name)	To designate a key as a primary key, FairCom supports the UNIQUE option.
CREATE VIEW	CREATE VIEW vw_sal (v_col1, v_col2) AS SELECT emp_id, name FROM sal	The column list is optional.
DROP INDEX	DROP INDEX emp.empuniq	
DROP VIEW	DROP VIEW vw_sal	[CASCADE   RESTRICT] is not supported.
SELECT	SELECT COUNT(emp_id), dept FROM mgrs GROUP BY dept HAVING dept > 15	In addition to supporting an order by on a column-list, as specified in the ODBC Programmer's Reference, FairCom has extended the syntax to support an order by on an expression-list or on any expression in a group by expression-list. For example: SELECT * FROM emp ORDER BY a+b,c+d,e This causes the result table to be ordered by three expressions: a+b, c+d, and e. If the expression is a positive integer literal, then that literal will be interpreted as the number of the column in the result set and ordering will be done on that column. No ordering is allowed on set functions or an expression that contains a set function.
subqueries	The following types of subqueries are supported: comparison, exists, quantified, in, and correlated. Order by clauses are not allowed in a subquery clause.	
approximate-numeric-literal	SELECT * FROM results WHERE quotient = -4.5E-2	
between-predicate	SELECT c1 FROM emp WHERE emp_id BETWEEN 10000 AND 20000	The syntax expr1 BETWEEN expr2 AND expr3 returns TRUE if expr1 >= expr2 and expr1 <= expr3. expr2 and expr3 may be dynamic parameters (e.g., SELECT * FROM emp WHERE emp_id BETWEEN ? AND ?).
correlation-name	SELECT * FROM emp t1, addr t2 WHERE t1.emp_id = t2.emp_id	FairCom supports both table and column correlation names.

Grammar	Examples	Comments
exact-numeric-literal	<pre>INSERT INTO cars (car_no, price) VALUES (49042, 49999.99)  SELECT * FROM numtbl WHERE c1 = -208.6543189</pre>	
in-predicate	<pre>SELECT * from colors WHERE color IN ('red', 'blue', 'green')</pre>	
set-function	<pre>SELECT COUNT(a+b) FROM q  SELECT MIN(salary) FROM emp</pre>	<p>MIN(expr), MAX(expr), AVG(expr), SUM(expr), COUNT(*), and COUNT(expr) are supported. COUNT(expr) counts all non-NULL values for an expression across a predicate. The following example counts all the rows in q where a+b does not equal NULL:</p> <pre>SELECT COUNT (a+b) FROM q</pre>
inner join syntax	<pre>SELECT * FROM mytableA, mytableB WHERE myColA = myColB  SELECT * FROM mytableA, mytableB INNER JOIN myColA = myColB</pre>	<p>These two statements are considered identical.</p>

## Extended SQL Grammar

The Extended Grammar supported by the c-tree ODBC Driver is as follows:

- Left Outer Join (two or three-table outer join)
- Unions
- Select
  - date arithmetic
  - date literal
  - time literal
  - timestamp literal
- extended predicates - (vendor string syntax)

The following tables provides some examples of the extended SQL grammar supported by the c-tree ODBC Driver:

Grammar	Examples	Comments
LEFT OUTER JOIN	<p>Two-table outer join:</p> <pre>SELECT * FROM emp LEFT OUTER JOIN dept ON emp.deptID = dept.deptID</pre> <p>Three-table outer join:</p> <pre>SELECT * FROM (emp LEFT OUTER JOIN dept ON emp.deptID = dept.deptID) LEFT OUTER JOIN addr ON emp.empID = addr.empID</pre> <p>Embedded in vendor strings:</p> <pre>SELECT t1.deptno, ename FROM {oj emp t2 LEFT OUTER JOIN dept t1 ON t2.deptno = t1.deptno}</pre>	<p>FairCom supports two-table outer joins.</p> <p>In addition to simple two-table outer joins, FairCom supports n- way nested outer joins.</p> <p>The outer join may or may not be embedded in a vendor string. If a vendor string is used, the ODBC driver will strip it off and parse the actual outer join text.</p>
UNION	<pre>SELECT name, status FROM tech_staff UNION SELECT name, status FROM adm_staff</pre>	UNION eliminates duplicate rows.
UNION ALL	<pre>SELECT name, status FROM tech_staff UNION ALL SELECT name, status FROM adm_staff</pre>	UNION ALL preserves duplicate rows.
date-literal	<pre>SELECT * FROM emp WHERE hire_date &lt; '1992-02-02'</pre> <pre>SELECT * FROM emp WHERE hire_date &lt; {d '1992-02-02'}</pre>	<p>FairCom supports the following date literal format: 'yyyy-mm-dd'.</p> <p>Dates may be in the range of year 0 to 9999.</p> <p>Date constants may be expressed in SQL statements as a character string or embedded in a vendor string. FairCom treats the character string representation as a string of type SQL_CHAR and the vendor string representation as a value of type SQL_DATE. This becomes important when conversions are attempted. For example, CONVERT({d '1992-02-02'}, SQL_TIMESTAMP) is valid, whereas CONVERT('1992-02-02', SQL_TIMESTAMP) returns an invalid SQL_TIMESTAMP value.</p>
time-literal	<pre>SELECT * FROM bday WHERE btime = '10:04:29'</pre> <pre>SELECT * FROM bday WHERE btime = {t '10:04:29'}</pre>	<p>FairCom supports the following time literal form: 'hh:mm:ss'.</p> <p>Time constants may be expressed in SQL statements as a character string or embedded in a vendor string. FairCom treats the character string representation as a string of type SQL_CHAR and the vendor string representation as a value of type SQL_TIME.</p>

Grammar	Examples	Comments
timestamp-literal	<pre>SELECT * FROM bday WHERE btime = '1965-08-25 05:25:00'</pre> <pre>SELECT * FROM bday WHERE btime={ts '1965-08-25 05:25:00'}</pre>	<p>FairCom supports the following timestamp literal format: 'yyyy-mm-dd hh:mm:ss'.</p> <p>Timestamp constants may be expressed in SQL statements as a character string or embedded in a vendor string. FairCom treats the character string representation as a string of type SQL_CHAR and the vendor string representation as a value of type SQL_TIMESTAMP.</p>
date arithmetic	<pre>SELECT * FROM inv WHERE inv_date &gt; '1993-01-01' AND inv_date &lt; {d '1993-01-01'} + 30</pre> <pre>SELECT * FROM pay WHERE pay_date - inv_date &gt; 30</pre>	<p>FairCom supports adding or subtracting an integer from a date where the integer is the number of days to add or subtract, and the date is embedded in a vendor string. (This is equivalent to executing a CONVERT on the date.)</p> <p>FairCom also supports subtracting one date from another to yield a number of days.</p>
extended predicates	{pred contains, col1, 'text'}	Uses extended vendor string syntax.

## 4.7 Driver Constraints

The following limits apply to the c-tree ODBC Driver. Addition limits may be imposed by third-party ODBC applications.

Description	Value
Concurrently open tables:	
16-bit multi-user non-server driver	305
All others	405
Number of rows:	2 billion
Number of columns in a query:	256
Size of a column:	2 gigabytes
Number of connections:	limited by memory
Table, index and column name length:	64 characters
Table qualifier:	68 characters
Maximum characters in a literal:	1000
Maximum columns in a CREATE TABLE statement:	256
Maximum ANDed predicates:	300
This example uses three ANDed predicates:	
<pre>SELECT * FROM abc WHERE c1 AND c2 AND c3 AND c4</pre>	

Description	Value
Number of joined tables:	limited by memory

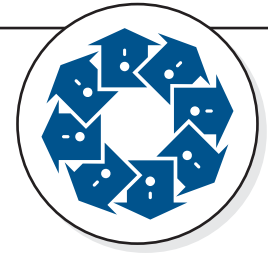
### Limitations of SQL\_LONGVARCHAR:

1. The LIKE predicate operates on the first 65500 characters of the column data.
2. All other predicates operate on the first 256 characters of the column data.
3. SELECT statements with GROUP BY, DISTINCT and ORDER BY return all of the data but only order on the first 256 characters of the column data.

## 4.8 c-tree ODBC Driver Licensing

The c-tree ODBC Driver is licensed for use on a single computer and is available in both 16-bit and 32-bit implementations (sold separately). For pricing information, please contact your software provider or nearest FairCom office.





## Advanced Topics

This chapter describes advanced topics, such as performance optimization, files installed, and Windows initialization file changes made during the install.

### 5.1 Performance Optimization

- The fewer file names in a given database, the faster the data access will be. Since the c-tree ODBC Driver can support multiple database names, it is usually advantageous to create multiple views into the database. See [“Multiple Database Script”](#) for further information.
- The fewer records per data file, the faster the data access will be.

### 5.2 Data Dictionary Creation - Import Method

The FairCom Data Dictionary can be created with the FairCom import utility, **import.exe** using a text-based script with the following layout:

```
DATABASE d_name
```

```
TABLE alias_name1(\path\file1_name1)
```

```
TABLE alias_name2(\path\file2_name2)
```

- *d\_name* is an arbitrary database name or view over the data files (i.e., *FAIRCOM.DB* is the database name used in the tutorial).
- *alias\_name1* and *alias\_name2* are arbitrary names for the data files residing on disk (i.e., *CUSTOMERS* and *ORDERS* are sample alias names from the tutorial import script, *SAMPLE.TXT*).
- *path1* and *path2* are either fully qualified paths or relative paths to the current logical drive (i.e., *FAIRCOM\ODBC\32bit\SAMPLE* from the tutorial import script, *SAMPLE.TXT*).
- *file\_name1* and *file\_name2* are the actual file names residing on disk (i.e., *CUSTMAST.DAT* and *CUSTORDR.DAT* from *SAMPLE.TXT*).

The following example shows one database view (*FAIRCOM.DB*) and four file members (*CUSTOMERS*, *ORDERS*, *ORDER\_DETAIL*, and *ITEMS*):

```
DATABASE FAIRCOM.DB
```

```
TABLE CUSTOMERS(c:\FAIRCOM\ODBC\32bit\sample\CUSTMAST.DAT)
```

```
TABLE ORDERS(c:\FAIRCOM\ODBC\32bit\sample\CUSTORDR.DAT)
```

```
TABLE ORDER_DETAIL(c:\FAIRCOM\ODBC\32bit\sample\ORDRITEM.DAT)
```

```
TABLE ITEMS(c:\FAIRCOM\ODBC\32bit\sample\ITEMMAST.DAT)
```

## Import Script

The import utility (**import.exe**) is a DOS-based, command line utility executed from the MS-DOS prompt. To create the dictionary for the ODBC tutorial, execute the import utility by specifying the path to **import.exe** located in the *FAIRCOM\ODBC\32bit* directory and pass the import script on the command line as follows:

```
C:\FAIRCOM\ODBC\32bit\import sample.txt
```

The Data Dictionary file, *CTSYSCAT.FCS*, is created in the root directory of your local drive. For example, if *FAIRCOM\ODBC\32bit* is installed on drive C, *CTSYSCAT.FCS* is created in the C:\ directory. Move this file to a new location by copying it to the desired location and specifying the path in the ODBC Setup dialog box. Each time the import utility is executed, any existing *CTSYSCAT.FCS* file in the root directory of the local drive is overwritten.

When building your own script, it is recommended to fully specify the file paths (i.e., include the logical drive letter in addition to the path, *C:\FAIRCOM\ODBC\32bit\*) as opposed to *\FAIRCOM\ODBC\32bit\*).

## Wildcard

Rather than building your own script, **import.exe** can be passed a wild card to build the script. Execute the import utility by specifying the path to **import.exe** located in the *FAIRCOM\ODBC\32bit* directory and pass *-f \*.dat* as follows:

```
C:\FAIRCOM\ODBC\32bit\import -f *.dat
```

Passing *-f \*.dat* tells **import.exe** to create the Data Dictionary (*CTSYSCAT.FCS*) and add all of the files with the ".dat" extension located in the current directory. When the import utility sees the *-f* switch, it also creates a text-based import file, *IMPORT.TXT*, listing all the data files located in the current directory. This script can be used by **import.exe** to re-create the dictionary.

## Multiple Database Script

As mentioned earlier, splitting application files into distinct database groups enhances performance of the ODBC Driver. This is easily done with **import.exe**.

Figure 5-2: Add New Database View over the Customer File

```
DATABASE FAIRCOM.DB

TABLE CUSTOMERS(c:\FAIRCOM\ODBC\32bit\sample\CUSTMAST.DAT)
TABLE ORDERS(c:\FAIRCOM\ODBC\32bit\sample\CUSTORDR.DAT)
TABLE ORDER_DETAIL(c:\FAIRCOM\ODBC\32bit\sample\ORDRITEM.DAT)
TABLE ITEMS(c:\FAIRCOM\ODBC\32bit\sample\ITEMMAST.DAT)

DATABASE CUSTOMER.DB

TABLE CUSTOMERS(c:\FAIRCOM\ODBC\32bit\sample\CUSTMAST.DAT)
```

The above illustrates adding a new database view (*CUSTOMER.DB*) over the Customer file (*CUSTMAST.DAT*). When this script is passed to **import.exe**, the data dictionary created contains two databases: *FAIRCOM.DB* with four data files; and *CUSTOMER.DB* with one data file. To switch between the databases, change the name specified in the Script Name setting in the ODBC Setup dialog.

## 5.3 File Locations

The c-tree ODBC installation process places several files on your system. The Microsoft and FairCom supplied files are installed in the Windows or \windows\system directories. The c-tree Sample data files are installed in the FairCom created *FAIRCOM\ODBC\32bit* directory or the directory specified during installation.

## 32-bit Microsoft Supplied Files

<i>ODBC32.DLL</i>	32-bit thunking Driver Manager
<i>ODBCCP32.CPL</i>	ODBC C Panel
<i>ODBCINT.DLL</i>	Language DLL
<i>ODBCAD32.EXE</i>	32-bit Administrator program
<i>ODBCCP32.DLL</i>	32-bit installer DLL
<i>ODBCCR32.DLL</i>	32-bit Cursor library
<i>ODBCINST.HLP</i>	Installer help file
<i>ODBCINST.CNT</i>	Installer help file table of contents
<i>CTL3D95.DLL</i>	Windows 95 three dimensional style library
<i>CTL3DNT.DLL</i>	Windows NT three dimensional style library
<i>MSVCRT20.DLL</i>	C run-time library

## 32-bit FairCom Supplied Files

<i>OTODBC32.DLL</i>	c-tree 32-bit ODBC Driver
<i>OTTREE32.DLL</i>	FairCom 32-bit c-tree Plus interface
<i>SIMSPY32.DLL</i>	ODBC Driver spy debug DLL
<i>OT_FPG32.DLL</i>	FairCom 32-bit c-tree Plus multi-user non-server DLL
<i>OT_TCP32.DLL</i>	FairCom 32-bit c-tree Plus TCP/IP client side DLL
<i>OT_NET32.DLL</i>	FairCom 32-bit c-tree Plus NETBIOS client side DLL
<i>OTCNFG32.DLL</i>	c-tree 32-bit configuration DLL
<i>fcodbc.hlp,</i> <i>fcodbc.cnt,</i> <i>roboex32.dll</i>	c-tree ODBC Driver Help files.

## FairCom Supplied Sample Files

The following FairCom sample files are for both the 16- and 32-bit implementations of the c-tree ODBC Driver.

<i>FAIRCOM.DB</i>	Script for tutorial, creates dictionary for sample files
<i>SAMPLE.TXT</i>	Script for creating the tutorial data dictionary
<i>IMPORT.EXE</i>	FairCom Data Dictionary creation utility
<i>CUSTMAST.DAT</i>	c-tree ODBC Tutorial customer master data file.
<i>CUSTMAST.IDX</i>	c-tree ODBC Tutorial customer master index file.
<i>CUSTORDR.DAT</i>	c-tree ODBC Tutorial customer order data file.

<i>CUSTORDR.IDX</i>	c-tree ODBC Tutorial customer order index file.
<i>ITEMMAST.DAT</i>	c-tree ODBC Tutorial item master data file.
<i>ITEMMAST.IDX</i>	c-tree ODBC Tutorial item master index file.
<i>ORDRITEM.DAT</i>	c-tree ODBC Tutorial order item data file.
<i>ORDRITEM.IDX</i>	c-tree ODBC Tutorial order item index file.

## 5.4 Advanced - Registry Options

The c-tree Drivers set several Registry keys. Use the [ConfigDSN\(\)](#) function to programmatically configure the c-tree ODBC Driver Data Source. Most c-tree Driver subkeys correspond to configuration entries shown in the table below. The other entries in this section are only adjustable in the Registry. All the subkeys below are in the key `\HKEY_CURRENT_USER\Software\ODBC\ODBC.INI\<name>`, where *<name>* defaults to one of the following:

- FairCom c-tree Driver for the c-tree V4 driver
- FairCom 32bit Driver for the 32bit c-tree Plus driver
- FairCom 16bit Driver for the 16bit c-tree Plus driver

<b>Registry subkey</b>	<b>Configuration option</b>
Alignment	Alignment
AllowUpdates	Allow Updates
CaseSensitive	Case-insensitive string Comparison
ColumnsOrderedBy	Column Order
CommEncryption	Use FETCPIP.DLL if Yes.
DataBuffers	Data Buffer Size
DataDictionary	Script Name
DBQ	Data Dictionary Path
DebugCtree	c-tree Plus Debug
DebugIndex	Debug Indexes
Description	Description
EnableFPUExceptions	Not a configuration option. See below.
Files	Number of Files
ForceClose	Not a configuration option. See below.
GuestLogin	Logon to Server as Guest
IndexBuffers	Index Buffer Size
KanjiConvert	Not a configuration option. See below.
MaxColSupport	Table's max number of columns

Registry subkey	Configuration option
OEMtoANSI	OEM to ANSI Data Conversion
Protocol	Driver Type
Sectors	Sector Size
ServerName	Server Name
SpecialTypes	Special Data Type Conversion DLL Name
StringDataPadding	Data Padding
StringKeyPadding	Key Padding

It is also possible to use these keywords in a connection string that uses a DSN. The effect is to override the particular settings that are specified. For example: "DSN=FairCom 32bit Driver;Alignment=4;" uses the settings from the registry for the data source 'FairCom 32bit Driver' and forces the alignment setting to 4.

**Note:** Three registry settings (ForceClose, KanjiConvert, and EnableFPUExceptions) are not supported in the command string.

## Adding a Shareable DSN

The Registry feature allows the creation and use of shareable file DSNs by the 32bit c-tree ODBC Driver. The Driver creates unshareable file DSNs automatically when creating user or system DSNs. To create a shareable file DSN:

1. Start the 32bit ODBC Administrator.
2. Select the **File DSN** tab.
3. Click the **Add...** button.
4. Select the ODBC driver for which you wish to create a file DSN (either "FairCom 32bit ODBC Driver" or "FairCom c-tree Driver").
5. Click the **Advanced...** button and enter the desired driver-specific keywords (from the list above). Both DRIVER= (driver name) and DBQ= (data dictionary path) are required. For example:
 

```
DRIVER={FairCom 32bit ODBC Driver}
DBQ=c:\faircom\odbc\32bit
```
6. Click the **OK** button.
7. Click the **Next** button.
8. Enter the name of your new file DSN.
9. Click the **Next** button.
10. Click the **Finish** button.

If after you follow the above steps, the ODBC administrator returns the error "A connection could not be made using the file data source parameters entered. Save non-verified file DSN?", you probably omitted needed parameters or specified invalid parameter values (see step 5 above). Click **Cancel** and start over at step 1.

After a file DSN is successfully created in this manner, it can't be configured from the ODBC Administrator, however, it is a plain text file that can be edited to change its settings.

This type of file DSN is known as a shareable file DSN - if you arrange for it to be shared by multiple machines (each with the c-tree ODBC Driver installed), it will serve as an ODBC data source that is identically configured on all of these machines.

## Force Close Option

For performance reasons, the c-tree ODBC Driver keeps data and index files open until the database is changed or the connection is closed. It is possible to force the Driver close files when the SQL engine requests files to be closed. To do so, create a registry key named ForceClose, as shown above, and set its value to "Yes" to enable this option.

## Japanese two-byte character (EUC) support

Our Japanese customers requested that the c-tree Plus TCP/IP client code be able to support Japanese two-byte character (EUC) encoding. c-tree Plus supports this option when #define ctJAPANESE is enabled. The c-tree ODBC Driver - c-tree Plus Edition now supports this feature when the following key is added to the registry:

```
HKEY_CURRENT_USER\Software\ODBC\ODBC.INI\FairCom 32bit Driver\KanjiConvert=EUC
```

Specifying any value other than "EUC" disables this feature.

## Optionally Disable Floating Point Unit Exceptions

By default, the c-tree ODBC Driver enables the divide by zero FPU (floating point unit) exception so that it can detect when a divide by zero error occurs. A customer requested the ability to avoid enabling this exception to work around a problem with Microsoft's .Net Font class constructor, which raises an ArithmeticException exception if floating point unit exceptions have been enabled.

The c-tree ODBC Driver checks the registry for a string value named EnableFPUEXceptions under the registry subkey:

```
HKEY_CURRENT_USER\Software\ODBC\ODBC.INI\FairCom 32bit Driver
```

If this string value exists and is set to No, the c-tree ODBC Driver does not enable FPU exceptions.

## 5.5 DSN-less ODBC connections

The c-tree ODBC Driver can be used without a DSN (data source name) by specifying the ODBC Driver Name rather than a DSN in the connection string, along with the desired options:

```
DRIVER=FairCom X ODBC Driver;keyword1=value1;...;keywordN=valueN;
```

The DRIVER specification above must be either "FairCom 32bit ODBC Driver" (for the c-tree Plus Driver) or "FairCom c-tree ODBC Driver" (for the c-tree V4 Driver). The Driver supports the following option keywords in the connection string.

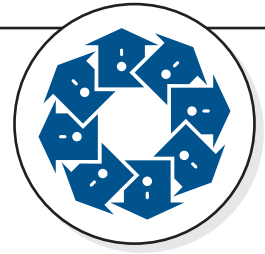
Keyword	Supported Values (default value listed first)
Alignment	Default, 1, 2, 4, 8
AllowUpdates	No, Yes
CaseSensitive	Yes, No
ColumnOrder	Position, Name
CommEncryption	No, Yes

<b>Keyword</b>	<b>Supported Values (default value listed first)</b>
DataBuffers	64
DataDictionary	FAIRCOM.DB
DBQ	c:\otree
DebugCtree	No, Yes
DebugIndex	No, Yes
Files	395
GuestLogin	Yes, No
IndexBuffers	64
MaxColSupport	256
OEMtoANSI	No, Yes
Protocol	MultiUser, TCPIP, NETBIOS, SPX
Sectors	16
ServerName	FAIRCOMS
SpecialTypes	None
StringDataPad	Zeros, Spaces, NulTermSpaces
StringKeyPad	Default, Zeros, Spaces

These keywords correspond to the registry settings described in [“Advanced - Registry Options”](#), with the following exceptions:

- ColumnOrder corresponds to the ColumnsOrderedBy registry setting.
- StringDataPad corresponds to the StringDataPadding registry setting.
- StringKeyPad corresponds to the StringKeyPadding registry setting.
- There are no supported connection string keywords for the ForceClose, KanjiConvert, and EnableFPUExceptions registry settings.
- The MaxRows setting can be used to limit the number of accessible rows, though there is no corresponding registry setting.





## Errors

### 6.1 DBOPEN DICTDBGETBYNAME(FAIRCOM.DB)=101

The specified database, *FAIRCOM.DB* in this message, could not be located. This error message is typically seen if Debug is enabled in the c-tree ODBC Setup dialog box.

1. The Data Dictionary Path was not properly specified. See [“Data Dictionary Creation - Import Method”](#) for an example.
2. The FairCom Data Dictionary has not been created.

### 6.2 No such database

1. If the FairCom Data Dictionary has been created, ensure the Data Dictionary Path in the ODBC Setup dialog box points to *CTSYSCAT.FCS*.
2. If the path is correct, re-create the FairCom Data Dictionary (*CTSYSCAT.FCS*).

### 6.3 No table names appear in the Select Table dialog box

Be sure the table name paths specified in the dictionary create script (i.e., *FAIRCOM.DB* for the tutorial) or the import script (i.e., *SAMPLE.TXT* for the tutorial) point to where the tables (files) are located.

### 6.4 Not able to open \*.FCS file

1. *CTSYSCAT.FCS* not created. Create the Data Dictionary.
2. *CTSYSCAT.FCS* not in the proper path. This file must reside in Data Dictionary Path as specified in the c-tree ODBC setup window.

### 6.5 Specified driver could not be loaded

Possible solutions:

1. Driver DLL may be missing. Reinstall the Driver.
2. Driver DLL may be corrupt. Reinstall the Driver.

3. Driver DLL not in \windows\system directory. Place Driver DLL in the *windows\system* directory.

## 6.6 Update failed. Continue editing record?

This error indicates an update failed. Possible causes are:

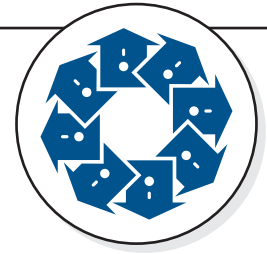
1. The record is in use by another user. Try the update later.
2. You have tried to add a duplicate data value to an index that does not support duplicates (i.e., the index is unique).
3. The required index field was not filled in prior to the insert request.

## 6.7 System Catalog's Files Need to be Rebuilt

Click **OK** to delete and recreate *CTSYSCAT.FCS* using the data dictionary script specified in the c-tree ODBC Driver setup dialog box. Otherwise, *CTSYSCAT.FCS* will not be touched and the connection attempt will fail.

## 6.8 Value Required

The c-tree ODBC Driver will not allow a NULL value for a field whose name given in the *DODA* begins with an asterisk (\*), or a field created using the NOT NULL column constraint. Put a non-null value in the necessary fields.



## Glossary

The following terms are used throughout this guide.

*16-bit*

An application designed to operate on a 16-bit operating system, such as Windows 3.x. Most 16-bit applications will also operate on 32-bit operating systems such as Windows 95/98 or Windows NT/2000.

*32-bit*

An application designed to operate only on 32-bit operating systems, such as Windows 95/98 and Windows NT/2000. The theory of 32-bit applications is that they are typically superior since they utilize the underlying power of the newer operating systems and hardware available today.

*c-tree®*

FairCom's legacy data engine. The parent product for c-tree Plus.

*c-tree Plus®*

FairCom's data engine. Used in many applications and embedded systems to manage data.

*column*

Synonymous with the definition of field. In the relational model, column is used in place of the term field. Column and field are used interchangeably in this guide.

*CTSYSCAT.FCS*

The FairCom Data Dictionary file.

*data dictionary*

A list of the data files used in various databases.

*directory*

A location where files are stored on disk. A directory can be thought of as a drawer in a file cabinet. Each file folder within the drawer can be thought of as a separate file, or collection of like information.

*FAIRCOM.DB*

The default script file used to create the FairCom Data Dictionary.

*field*

A specific piece of information stored within a record. Many fields are commonly stored in one record. For example, a record storing a customer address might contain fields for the customer's name, address, city, state, zip, and phone number. This example therefore contains 6 fields in each record.

*file*

A collection of like information, referred to as records. See the definitions for directory, record and table for further information.

*index file*

A special type of file that provides a mechanism for performing fast data retrieval.

*record*

A piece of information stored within a data file. Expanding on the file cabinet example used in the directory definition, each piece of paper found within a file folder can be thought of as a record. A record is a unique piece of information similar to other pieces of information (papers) within the file folder.

*row*

Synonymous with definition of record. In the relational model, row is used in place of the term record. Row and record are used interchangeably in this guide.

*segment*

A piece of information (field) stored within an index file. An index can be made up of many fields or even portions of fields. Each field or portion of a field stored within an index is called a segment.

*table*

A term synonymous with the definition of file. In the relational model, table is used in place of file. Table and file are used interchangeably in this guide.

*unique index*

An index that does not accept duplicate records, indicating each value stored in the index is unique. An index defined over a customer social security number field would typically be a unique index. If the index supports duplicate records, it is said to be non-unique or duplicate allowed. An index defined over one field, last name, would most likely accept duplicate records since it may be common to have more than one customer with the last name of Smith.

*VENDOR.DB*

A script file used to build the data dictionary for a vendor's application data files.

# Index

<b>3</b>	
32-bit FairCom Supplied Files.....	33
32-bit Microsoft Supplied Files.....	33
<b>A</b>	
Add	
files.....	18
shareable DSN.....	35
Adding a Shareable DSN.....	35
Adding Files to the Dictionary.....	18
Additional tips for the ctree Server Script.....	11, 12
Advanced - Registry Options.....	34, 37
Advanced topics	
create data dictionary.....	31
DSN-less ODBC connections.....	36
file locations.....	32
import method.....	31
performance optimization.....	31
registry options.....	34
Advanced Topics.....	9, 31
<b>B</b>	
Browse method.....	10
Browse Method.....	9, 10
Building a FairCom Data Dictionary Script ..	2, 14, 17
<b>C</b>	
Configuration.....	3
Connections	
DSN-less.....	36
Constraints.....	28
Conversion	
process.....	19
Copy sample data.....	2
Copy Sample Data.....	2
Core SQL Grammar.....	24
Create	
data dictionary.....	2
multiple database references.....	18
c-tree ODBC Driver Licensing.....	29
c-tree ODBC Driver Purpose.....	19
c-tree ODBC Driver Setup.....	3
c-tree ODBC Driver Technical Details.....	19
c-tree Server Tutorial Preparation.....	2
<b>D</b>	
Data	
copy sample.....	2
source name.....	3
Data dictionary.....	17
adding files.....	18
automatic creation.....	11
create.....	2, 31
multiple database.....	17
path error.....	39
Data Dictionary Creation - Import Method .	12, 31, 39
Database	
create multiple references.....	18
multiple.....	17
multiple script.....	32
script name.....	3
DBOPEN	
DICTDBGETBYNAME(FAIRCOM.DB)=101.....	39
Defining Multiple Databases.....	17
Disable floating point unit exception.....	36
Driver	
compliance.....	21
configuration.....	3
constraints.....	28
description.....	3
licensing.....	29
purpose.....	19
requirements.....	20
SQL conformance.....	23
type.....	3
types.....	19
Driver Constraints.....	28
Driver type	
developer-only.....	19
end-user.....	19
end-user read-only.....	19
Driver Types.....	19
DSN-less ODBC connections.....	36
<b>E</b>	
Errors.....	39
database not located.....	39
driver not loaded.....	39
no such database.....	39
no table name.....	39
not able to open .fcs file.....	39
rebuild system catalog file.....	40
update failed.....	40
value required.....	40
Extended SQL Grammar.....	26
<b>F</b>	
FairCom Data Dictionary - In Depth.....	14, 17
FairCom Supplied Sample Files.....	33
FAIRCOM TYPOGRAPHICAL CONVENTIONS.....	v
File Locations.....	32
Files	
add.....	18
locations.....	32
system catalaog error.....	40
Floating point unit exceptions	
disable.....	36
Force close option.....	36
Force Close Option.....	36

<b>G</b>	
Generic ODBC Compliant Application Notes.....	18
Getting Started with Your Data .....	9
Glossary .....	41
<b>I</b>	
Import	
method .....	31
script.....	32
Import Script.....	32
Installation .....	1
driver .....	1
vendor supplied script .....	9
Installing a Vendor Supplied Script.....	9
<b>J</b>	
Japanese character support .....	36
Japanese two-byte character (EUC) support.....	36
<b>L</b>	
Limitations .....	28
SQL_LONGVARCHAR .....	29
Limitations of SQL_LONGVARCHAR:.....	29
<b>M</b>	
Minimum SQL Grammar .....	23
Multiple database .....	17
Multiple Database Script .....	31, 32
<b>N</b>	
No such database .....	39
No table names appear in the Select Table dialog box .....	39
Not able to open *.FCS file .....	39
<b>O</b>	
ODBC Compliance.....	21
ODBC History/Overview .....	1
ODBC Setup Dialog Options.....	14
ODBC/Application Setup.....	5, 9, 13
Optimize	
performance .....	31
Optionally Disable Floating Point Unit Exceptions.	36
Options	
force close.....	36
registry .....	34
Options>>.....	15
Overview .....	1
<b>P</b>	
Performance optimization .....	31
Performance Optimization .....	31
<b>Q</b>	
Quick Start .....	1
<b>R</b>	
Registry options .....	34
Requirements.....	20

<b>S</b>	
Scalar Functions .....	21
Script	
import .....	32
method .....	11
multiple database .....	32
tips.....	12
Script Method .....	9, 11
Script name .....	3
Setup	
dialog options .....	14
driver .....	3
Shareable DSN	
add .....	35
Specified driver could not be loaded.....	39
SQL .....	23
core grammar.....	24
extended grammar .....	26
minimum grammar .....	23
SQL Conformance .....	23
System Catalog_s Files Need to be Rebuilt .....	40
<b>T</b>	
Tutorial .....	4
copy sample data .....	2
create data dictionary.....	2
driver setup.....	3
file layout .....	6
preparation .....	2
setup.....	1
Tutorial File Layout.....	6
Tutorial Setup .....	1
<b>U</b>	
Update failed. Continue editing record?.....	40
<b>V</b>	
Value Required .....	40
<b>W</b>	
Wildcard .....	32