DataLens Help Contents

About DataLens

DataLens lets you read data from and write data to external tables without leaving 1-2-3 for Windows. The external table can be on a personal computer, a network server, a corporate mainframe, or a CD-ROM.

DataLens is integrated with 1-2-3 so you do not need to learn new, complicated programs and commands. You work with familiar commands and menu choices to use or create external data when you need to and to generate your own database queries.

DataLens gives you this flexibility by using a driver to communicate between your application and the data source, as shown in the figure below. If the driver communicates directly with a database management system (DBMS), such as the DataLens driver for SQL Server, then the data source is the DBMS. If the driver manipulates data in external tables without going through a DBMS, such as the DataLens driver for dBASE IV tables, then the data source is an external database table in a particular file format. The DataLens driver sends commands your application to the external table and returns results to 1-2-3.

DataLens Capabilites

With DataLens, you can use data in external tables from 1-2-3 without learning new tools or data query languages. Depending on the specific features of the DataLens driver you are using, you can do the following:

- Create a table
- Request data from a table
- Add information to a table
- Update or delete information already in a table
- Select specific records from a table
- Perform calculations on information in a table

Using the DataLens Driver for dBase IV Tables

The DataLens driver for dBASE IV tables is compatible with dBASE III® and IV (Version 1.5 or lower) database and index files, and FoxPro® 2.0 database files that contain no memo or picture fields. This section describes the capabilities of the dBASE IV driver and provides information you should know when you perform any operation on external dBASE IV tables.

> **Note** Although this topic refers to dBASE IV tables, the information also applies to dBASE III Plus® tables, unless otherwise noted.
>
> **Note** Before using the dBASE driver, you must run DOS SHARE before starting Windows.

Using the 1-2-3 sample file

Your application provides a 1-2-3 sample file with each DataLens driver. You use the file to create an external table. See Using a 1-2-3 model range for more information. You can then use the external table with the procedures in Chapter 21 in the *User's Guide*.

The sample file for the dBASE IV driver, EMPLOYEE.DBF, is in the subdirectory SAMPLE\dbase in your 1-2-3 directory. If you are using the network version of 1-2-3, SAMPLE\dbase is a directory in your personal directory. Before using the sample dBASE IV file, save a copy of the file in a different directory, in case you want to use the original version of the file again.

Capabilities of the dBASE IV driver

By using the dBASE IV driver with 1-2-3, you can access, modify, create, and delete external dBASE IV tables from within 1-2-3. To use the 1-2-3 Tools Database commands, Query commands, and database @functions with external tables, you specify the range name you assigned to the dBASE IV table as the database table.

Basic database tasks

The table below lists the basic database tasks you can perform on external tables using the dBASE IV driver, and the 1-2-3 commands you use to accomplish those tasks after you connect to an external table.

Database task	Description	1-2-3 commands
Activate indexes	Activate indexes for external tables	Tools Database Send Command
Create tables	Create external tables	Tools Database Create Table
Delete rows	Delete records from external tables	Tools Database Delete Records

Delete tables	Delete external tables	/Data External Delete
Extract data	Copy records or specific fields from an external table to 1-2-3	Tools Database New Query
Insert rows	Add records to external tables	Tools Database Append Records
Update rows	Edit records in external tables	Query Update Database Table

In addition, you can use database @functions to perform calculations in 1-2-3 with data that resides in external tables. You can also use Tools Database Send Command to send commands to the driver.

> **Note** Deletion of records works as it does in dBASE; the records are marked for deletion but not actually removed from the dBASE table, although they will not appear. To actually remove records from the database table, use the {SEND-SQL} macro command to send a PACK command to the database. Alternatively, you can use PACK in the dBASE environment.
>
> **Help** For information about using @functions with databases, search on “database @functions” in Help. For information on using macros with external databases, search on “external database tables” in main Help for 1-2-3 and select the topic “External Table Control Macro Commands Category.”

Using dBASE IV index files

When using the dBASE IV driver and Tools Database Send Command, you can tell the dBASE IV driver which index files are associated with the database or which index file to use for the current sort order. dBASE IV uses index files to change the sort order it displays for records in tables. You can use the dBASE IV driver to access and update tables that use multiple index files (.MDX files) and single index files (.NDX files). An .MDX file can contain up to 47 indexes, each with its own name, called a tag.

> **Note** dBASE III Plus supports only .NDX files, not .MDX files.

The dBASE IV driver automatically updates all tags in an .MDX file when you modify the table associated with the .MDX file. It also updates all .NDX files that have been activated by Tools Database Send Command. When extracting data from a table, you first use a tag in the .MDX file or use an .NDX file to retrieve data in the sequence of that index or tag. Using an index file is an efficient way to extract your data only if you have specified criteria that are supported by the index.

If your criteria select less than 20% of the records in the database table, for example, an index generally speeds retrieval. However, if your criteria select most of the records in a database, using an index slows retrieval. It may be faster to extract the records without an index and sort them using Query Sort.

To use a tag from the .MDX file, choose Tools Database Send Command and enter the following command string:

set order to tag [*tag name*] table [*range name*]

where *tag name* is the name of the tag you want to use and *range name* is the name of the table you are using.

The dBASE IV driver updates an .NDX file only if you have opened the file. You can open several .NDX files at one time, depending on the setting of the FILES parameter in your CONFIG.SYS file. To open an .NDX file from 1-2-3, choose Tools Database Send Command and enter the following command string:

set index to file [*file name*] table [*range name*]

where *file name* is the name of the .NDX file and *range name* is the name of the table you want to use with the index. Do not enter file extensions in the command. (If you do not open the .NDX file, you can use REINDEX or PACK in dBASE IV to update the index.)

After specifying the index file to use, when you extract data from a table, the records are sorted in the order specified by the index file.

You can have multiple indexes open per table. To change the index or tag you want to use with a table, you must close any open indexes or tags, then open the new index. To close an index or a tag, choose Tools Database Send Command and enter the following command string:

close index table [*range name*]

where *range name* is the name of the table you are using.

Driver-supported dBASE IV files and commands

The dBASE IV driver supports only .DBF, .DBT, .MDX, and .NDX files. It does not use or update other types of dBASE IV files. In addition, when you use 1-2-3 to delete a table, the dBASE IV driver deletes the .DBF, .DBT, and .MDX files; it does not delete any associated files, such as forms, reports, and index files.

By using the dBASE IV driver with Tools Database Send Command or the {SEND-SQL} macro command, you can send SQL commands directly to the database. For a list of keywords and option values that the dBASE IV driver supports, see Database and driver communication and security or see your database administrator.

Help For more information about the keywords and option values, search on "dBASE" in Help for 1-2-3.

Creating dBASE IV tables

With the dBASE IV driver, you can use 1-2-3 to create dBASE IV tables. To create a dBASE IV table, you can use an existing table or a 1-2-3 database table as a model and then use Tools Database Create Table, or you can create a table definition and use /Data External Create from the 1-2-3 Classic® menu.

This section provides information about using a model, and describes the table definition for dBASE IV tables. Use this information with the procedures in Chapter 22 in the *User's Guide*.

Using a 1-2-3 model range

If you use a model range to create a table, the driver uses default data types when creating the table. The table below shows 1-2-3 data types and the default data types the driver converts them to when it creates a dBASE IV table.

1-2-3 data type	Data type in dBASE IV table
Label	Character
Value	Numeric
Value formatted as a date	Date

When you create an external table using a model, 1-2-3 prompts you to enter a creation command. You can use the creation command to specify parameters such as LOCK MODE.

To specify non-default data types, you create a table definition and use /Data External Create from the 1-2-3 Classic menu to create an external table.

About the table definition

The table definition contains information that tells the driver how to set up the dBASE IV table. A table definition contains six columns of information, although the dBASE IV driver only uses information from columns 1, 2, 3, and 6, which are described below.

- Column 1 contains field names. This information is required for each field in the table.

  Field names must be unique. Because dBASE IV is not case-sensitive, it considers uppercase and lowercase versions of the same character to be the same. For example, "NAME" and "Name" are considered to be the same field name.

  Field names can contain as many as 10 characters, which can include letters, numbers, and _ (underscores). The first character, however, must be a letter.

  When using the dBASE IV driver, the maximum number of fields in an external table is 255.

- Column 2 contains data types. This information is required for each field in the table. For information about data types, see About dBASE IV data types.

- Column 3 contains field widths. An entry in this column is required for character fields.

  The field width indicates the maximum width of a field. For Numeric fields, you can also specify the number of decimal places. The driver ignores field widths that you enter in the table definition for date fields, which always have a width of 8; logical fields, which have a width of 1; and memo fields which have a length of 10. The maximum combined width of all fields in a record is 4,000 characters.

- Column 6 contains field creation strings. It may also be used to designate decimal places for float fields. For details, see “About float fields” below.

The table below shows a table definition for a dBASE IV table. Although the columns for *Column labels* and *Field descriptions* must be present in the table definition, the dBASE IV driver ignores entries in these columns because they do not apply to dBASE IV tables.

Field names	**Data types**	**Field widths**	**Column labels**	**Field descriptions**	**Field creation strings**
EMPID	Numeric	11,2	NA	NA	NA
LAST	Character	12	NA	NA	NA
FIRST	Character	12	NA	NA	NA
DOH	Date	8	NA	NA	NA
SALARIED	Logical	1	NA	NA	NA
DEPTNUM	Numeric	5,2	NA	NA	NA
SALES	Float	9	NA	NA	dec=2
COMMENT	Memo	10	NA	NA	NA

About dBASE IV data types

Databases let you use many different types of data. It is important for you to know the data types dBASE IV supports when you create new tables (as well as when you enter and analyze data). 1-2-3 uses these data types in table definitions when you use /Data External Create.

When you send data between dBASE IV tables and 1-2-3, the dBASE IV driver converts dBASE IV data types to 1-2-3 data types, and vice versa, so that both programs can use and understand the same data. The table below shows dBASE IV data types and the corresponding 1-2-3 data types. The table also shows the field widths and the values allowed for the various data types.

dBASE IV data type	**Field width**	**Values accepted by dBASE IV**	**Data type in 1-2-3**
Character	1-254	Character strings containing letters, numbers, and special characters	Label
Date	8	Date values from January 1, 1900 through December 31, 9999	Value (date number)
Float	1-20	Numbers with up to 19 digits and 18 decimal places	Value
Logical	1	T, F, t, f, Y, N, y and n for true, false, yes and no	Value (1 or 0 only)
Memo	65,535	Up to 511 characters per record	Label
Numeric	1-20	Numbers with up to 19 digits and 18 decimal places	Value

About character fields

You can enter up to 254 characters in a character field, depending on the width you create for the field.

About date fields

When you extract date fields from dBASE IV tables, 1-2-3 enters the dates as date numbers in the worksheet and formats the cells with the date format specified in Tools User Setup International. Although 1-2-3 cannot display dates before January 1, 1900 or after December 31, 2099, it can store all dates from dBASE IV tables and return them to the tables. When you format cells that contain dates outside the range that 1-2-3 can display, 1-2-3 displays *** (asterisks) in the cells.

About float fields

When creating a table, the dBASE IV driver accepts numbers in float fields up to 19 digits and 18 decimal places. If no decimal places exist, you can create a 20-position field. If a number is outside of this range, the driver displays a message.

When you create a table definition, you specify the width as the number of digits excluding the minus sign and decimal places in column 3. To specify the number of decimal places for a float field, enter DEC=*xx* in column 6, where *xx* represents the number of decimal places you want the field to have. You can enter any number from 0 through 18, but the number of decimal places must be less than the field width in column 3.

About logical fields

1-2-3 displays logical values as 1 (for dBASE IV entries T, t, Y and y) and 0 (for dBASE IV entries F, f, N and n). When you use 1-2-3 to add or modify records in external tables, entering 1 in 1-2-3 enters T in the dBASE IV table. Entering 0 in 1-2-3 enters F in the dBASE IV table.

About memo fields

The dBASE IV driver can store a maximum of 511 characters in a memo field. If the field contains more than 511 characters, you cannot update the records. If this happens, retrieve all fields in the record into 1-2-3, modify the field so that it contains 511 characters or fewer, then update the original set of records.

About numeric fields

When creating a table, the dBASE IV driver accepts numbers in numeric fields up to 19 digits and 18 decimal places. If no decimal places exist, you can create a 20-position field. The driver displays a message if a number is outside of this range.

In column 3 of a table definition, the field width of a numeric field appears as two numbers separated by a , (comma), such as 11,2. The number preceding the comma represents the field width excluding the minus sign and decimal places; the number after the comma represents the number of decimal places. 11,2 indicates that a numeric field has 11 digits, 2 of which are to the right of the decimal. To enter a width and a number of decimal places in the table definition, you must precede the entry with a label prefix, such as ’11,2. The number of decimal places you specify must be less than the field width.

> **Caution** When you use negative entries in numeric fields, the - (minus sign) counts as an integer position. When you enter a minus sign and four numerals, for example, the field width is 5. If you enter a minus sign and five numerals, the rightmost numeral is truncated from the entry.
>
> **Note** When you extract records from dBASE IV tables that contain blank fields, the blank fields become empty cells in 1-2-3.
>
> **Note** The dBASE IV driver supports the following precision and scale of numbers:
>
> 0<precision<15; 0<scale<precision
>
> **Note** Scientific notation is not supported.
>
> **Note** Numbers must conform to the dBASE IV integer and decimal place definitions.

File Access

When sharing files on a network, with the default setting of Locking=File, you can use 1-2-3 to write to a table only if no one else is modifying the table. If a dBASE IV table is a read-only table, you can read the data in the table but you cannot write to the table.

Multiple 1-2-3 Release 5 users can access and read the same tables and indexes concurrently. If dBASE users need access to the same dBASE table that you are using, file level locking is required. For more information about locking, see For Administrators.

Using macros for transaction control

When you use a macro to modify an external table, the database management system does not actually modify the table until the driver commits (finalizes) the modification transaction. The driver commits a transaction when any of the following occurs:

- The macro containing a command that modifies the external table ends.
- A command that modifies the external table through the menus is executed.
- The macro uses the advanced macro command {COMMIT}.

When a command is waiting to be committed and the database administrator has set Locking=Record in the ODBC dBASE Driver Setup dialog box or in the ODBC.INI file, no 1-2-3 users can update those records, although they may update other records. If the setting is Locking=File (the default), users cannot modify any record in the file.

If the macro uses Tools Database Disconnect or the connection to the external table disconnects in some other way, the driver does not commit the command that is waiting but instead rolls back (deletes) the command.

To be sure macro transactions are committed, use {COMMIT} after each of the following commands, which insert and delete records, and update tables: Tools Database Append Records, Tools Database Delete Records, and Query Update Database Table.

{COMMIT} uses two optional arguments: {COMMIT *"drivername", "databasename"*}. If you use the optional arguments, the driver commits only the transaction pending for the driver and database you specify. Be sure to enclose the driver name and database name in " " (quotation marks). If you do not use the optional arguments, the driver commits all pending transactions.

Use the advanced macro command {ROLLBACK} to cancel a command in a macro that modifies an external table. To be effective, you must use {ROLLBACK} before the macro commits the command you want to roll back. {ROLLBACK} also uses two optional arguments: {ROLLBACK *"drivername", "databasename"*}.

Table security

No standard protection scheme exists for databases. The dBASE IV driver does not support passwords or user IDs. To use a dBASE IV table, you must be able to access the table without using a password or user ID. 1-2-3 supports locking at the file level to prevent data corruption.

The dBASE IV driver does not support encrypted files.

Supported character sets

The dBASE IV driver supports the following character sets: United States (code page 437), Multilingual (code page 850), Portuguese (code page 860), French-Canadian (code page 863), and Nordic (code page 865). The default character set is United States.

Although 1-2-3 stores characters from the character sets listed above, it displays and prints characters from the ANSI character set, which is the character set that Windows supports.

Changing character sets

Some drivers let you use more than one character set. If a table was created with a character set other than the character set 1-2-3 is using, you may have to use /Data External Other Translation from the 1-2-3 Classic menu to change the character set in order to read all the data from the table. For more information about using the 1-2-3 Classic menu, see Chapter 5 in the *User's Guide*.

To change a character set

1. If you have not already done so, use Tools Database Connect to External to connect to an external table in the database whose character set you want to change.
2. Choose /Data External Other Translation from the 1-2-3 Classic menu.

   1-2-3 prompts you to enter the name of the database whose character set you want to change and displays the names of DataLens drivers.
3. Enter the name of the driver associated with the database whose character set you want to change.

   1-2-3 displays the names of databases.

4. Enter the name of the database for which you want to select a different character set.
5. Select the character set you want to use.

The character set you select remains in use for all of the external tables in the external database you specified until you end the current work session or select another character set for that external database.

Using the DataLens Driver for Paradox Tables

The DataLens Driver for Paradox tables (the Paradox driver) is compatible with the table format of Paradox 3.5 and Paradox 4.0. This topic describes the capabilities of the Paradox driver and provides information you should know when you perform any operation on external Paradox tables from within 1-2-3.

Using the 1-2-3 sample file

1-2-3 includes a sample worksheet file with each DataLens driver. You use the file to create an external table. See Using a 1-2-3 model range for more information. You can then use the external table with the procedures in Chapter 21 in the *User's Guide*.

The sample file for the Paradox driver, EMPLOYEE.DB, is in the subdirectory SAMPLE\DBASE in your 1-2-3 directory. If you are using the network version of 1-2-3, SAMPLE\DBASE is a subdirectory in your personal directory.

Capabilities of the Paradox driver

By using the Paradox driver with 1-2-3, you can access, modify, create, and delete external Paradox tables from within 1-2-3. To use the 1-2-3 Tools Database commands, Query commands, and database @functions with external tables, you specify the range name you assigned to the Paradox table as the database table.

Basic database tasks

The table below lists the basic database tasks you can perform on external tables using the Paradox driver, and the 1-2-3 commands you use to accomplish those tasks after you connect to an external table.

Database task	Description	1-2-3 commands
Create tables	Create external tables	Tools Database Create Table
Delete rows	Delete records from external tables	Tools Database Delete Records
Delete tables	Delete external tables	/Data External Delete
Extract data	Copy records or specific fields from an external table to 1-2-3	Tools Database New Query
Insert rows	Add records to external tables	Tools Database Append Records
Update rows	Edit records in external tables	Query Update Database Table

You can use database @functions to perform calculations in 1-2-3 with data that resides in external tables. You can also use Tools Database Send Command to send commands to the driver.

Help Help includes syntax, arguments, notes and examples for every 1-2-3 @function. For more information about 1-2-3 database @functions, search on “Database @functions” in main Help for 1-2-3.

Driver-supported Paradox commands

Because the Paradox driver does not use Paradox to access the external tables, you cannot use Paradox commands from within 1-2-3.

Driver-supported Paradox files

The Paradox driver supports only .DB and .PX files. It does not use or update other types of Paradox files. When you use 1-2-3 to delete a table, the Paradox driver also deletes the associated primary index (.PX) file, if it exists. The driver does not delete other associated files, such as forms, reports, scripts, or secondary index files.

Creating Paradox tables

With the Paradox driver, you can use 1-2-3 to create Paradox tables. To create a Paradox table, you can use an existing external table or a 1-2-3 database table as a model and then use Tools Database Create Table, or you can create a table definition and use /Data External Create from the 1-2-3 Classic menu.

This section provides information about using a model, gives guidelines for using an index, describes the table definition for Paradox tables, and discusses sort orders.

Using a 1-2-3 model range

When you use a model range to create a table, the driver uses the default data types when creating the table. In addition, you cannot specify index fields when you create a table from a model range.

The table below shows 1-2-3 data types and the default data types the driver converts them to when it creates a Paradox table.

1-2-3 data type	Data type in Paradox table
Label	Alphanumeric
Value	Number
Value formatted as a date	Date

To specify non-default data types, you create a table definition and use /Data External Create from the 1-2-3 Classic menu to create an external table.

Using an index with the Paradox driver

When you use or create tables with index files, keep the following points in mind:

- You can use the Paradox driver to create, access, update, and delete tables with primary indexes. To create an indexed table in 1-2-3, precede the data type of index fields with an * (asterisk), such as *Alphanumeric, in the table definition. List all the index fields first, in order of precedence. All of the index fields must appear before any non-index fields. All of the index fields combined comprise the primary index for the table.

  For more information about creating tables in 1-2-3, see Chapter 22 in the *User's Guide*.
- If you delete tables with primary indexes, the Paradox driver deletes the associated primary index (.PX) files, but does not delete other associated files, such as forms, graphs, reports, image settings, validity checks, or secondary index files.
- You can access and update information in tables with secondary indexes, but you cannot create tables with secondary indexes. If you use the Paradox driver to update information in tables with secondary indexes, Paradox updates the secondary indexes the next time you use the tables with Paradox. (The Paradox driver updates primary indexes.)
- The Paradox driver does not modify primary fields in an indexed record or insert records whose entries in the indexed fields are identical to entries in the indexed fields of existing records. To insert the new record with an identical index, you must delete the existing record.
- The index (.PX) file must be located in the same directory as the (.DB) table file.

About the table definition

The table definition contains information that tells the driver how to set up the Paradox table. A table definition contains six columns of information, although the Paradox driver only uses information from columns 1, 2, and 3, which are described below.

- Column 1 contains field names. This information is required for each field in the table.

  Field names must be unique. Because Paradox is not case-sensitive, it considers uppercase and lowercase versions of the same character to be the same. For example, “NAME” and “Name” are considered to be the same field name.

  Field names can contain as many as 25 characters.

  When using the Paradox driver, the maximum number of fields in an external table is 255.

- Column 2 contains data types. This information is required for each field in the table. For information about data types, see “About Paradox data types” below. For information about using data types to create an indexed table in 1-2-3, see Using an index with the Paradox driver.

- Column 3 contains field widths. An entry in this column is required for alphanumeric fields.

  The field width indicates the maximum width of a field. The driver uses field widths from the table definition for alphanumeric fields only. Do not enter field widths for other data types in the table definition.

The table below shows a table definition for a Paradox table. Although columns 4, 5, and 6 must be present in the table definition, the Paradox driver ignores entries in these columns because they do not apply to Paradox tables.

Field names	**Data types**	**Field widths**	**Column labels**	**Field descriptions**	**Field creation strings**
EMPID	Number	8	NA	NA	NA
LAST	Alphanumeric	12	NA	NA	NA
FIRST	Alphanumeric	12	NA	NA	NA
DOH	Date	4	NA	NA	NA
SALARIED	Number	8	NA	NA	NA
DEPTNUM	Number	30	NA	NA	NA

About Paradox data types

Databases let you use many different types of data. It is important for you to know the data types Paradox supports when you create new tables (as well as when you enter and analyze data). 1-2-3 uses these data types in table definitions when you use /Data External Create.

When you send data between 1-2-3 and Paradox tables, the Paradox driver converts Paradox data types to 1-2-3 data types, and vice versa, so both programs can use and understand the same data. The table below shows Paradox data types and the data types the driver converts them to in 1-2-3. The table also shows the field widths and the values allowed for the various data types.

Paradox data type	**Field width**	**Values accepted by Paradox**	**Data type in 1-2-3**
Alpha- numeric	1-255	Character strings containing letters, numbers, and special characters	Label
Number	8	±1.7E-308 through ±1.7E+308 with 15-digit precision	Value
Currency	8	Number entries with a currency display format	Value
Date	4	Dates from January 1, 1900 through December 31, 9999	Value (date number)
Short	2	Integers from -32,767 through 32,767	Value

Notes about Paradox data types

- The Paradox driver uses the field width in the table definition only when creating fields whose data type is alphanumeric. The driver ignores the field width for all other data types.
- When you extract date fields from Paradox tables, 1-2-3 enters the dates as date numbers in the worksheet and formats the cells with the date format specified in Tools User Setup International. Although 1-2-3 cannot display dates before January 1, 1900 or after December 31, 2099, it can store all dates from Paradox tables and return them to the tables. When you format cells that contain dates outside the range that 1-2-3 can display, 1-2-3 displays *** (asterisks) in the cells.
- Although 1-2-3 can only display numbers from ±1.0E-99 through ±9.99E+99, it can store all numbers from Paradox tables and return them to the tables. 1-2-3 displays *** (asterisks) in cells that contain numbers outside the range that 1-2-3 can display.
- Paradox does not support time entries from 1-2-3. If a 1-2-3 record includes an entry with a date portion and a time portion, when you enter that record in Paradox, Paradox maintains only the date portion of the entry.

Paradox sort orders

The Paradox driver lets you use table creation strings to specify a sort order for the table. Paradox uses alternative sort orders to allow users to sort records in indexed tables based on their native language. The Paradox driver supports the Paradox sort orders ASCII, Intl (International), SwedFin (Swedish and Finnish), and NorDan (Norwegian and Danish) for indexed tables. The Paradox driver maintains the index in the sort order that was used when the table was created. When creating a new indexed Paradox table, the driver uses the ASCII sort order unless you specify a different sort order in the table creation string when you use 1-2-3 to create a new table. The table below shows valid table creation strings.

Table creation string	Description
SORT ASCII	ASCII sort order
SORT INTL	International ASCII sort order
SORT NORDAN	Norwegian and Danish sort order
SORT SWEDFIN	Swedish and Finnish sort order

File access

When sharing files on a network, the following rules apply:

- You can write to a table only if no one else is using the table.
- Multiple users can read a table concurrently.
- You can use 1-2-3 to read a table when someone is using Paradox to co-edit the table.

If you are prompted for a user ID and password when connecting to the Paradox driver but do not need one to connect to the driver, choose OK or press enter without entering a user ID or password.

Table security

When using the Paradox driver, you can send commands to encrypt and decrypt your tables and to change the current password. To encrypt a table is to scramble the data in the table so that no one can read the table without knowing the correct password. To decrypt a table is to unscramble the data. The current password is the last password you entered while using Tools Database Connect to External.

To encrypt a table, choose Tools Database Send Command and enter the following command string:

ENCRYPT=*filename*[,*password*]

where *filename* is the name of the file you want to encrypt and *password* is an optional password you can enter for the table. If you do not enter a password, the driver uses the current password.

To decrypt a table, choose Tools Database Send Command and enter the following command string:

DECRYPT=*filename*[,*password*]

where *filename* is the name of the file you want to decrypt and *password* is the password you assigned to this table when you encrypted it. If you do not enter a password, the driver uses the current password.

To change the current password, choose Tools Database Send Command and enter the following command string:

PASSWORD=*password*

where *password* is the password you want to make current.

For more information about using Tools Database Send Command, see Chapter 22 in the *User's Guide*.

Supported character sets

The Paradox driver supports the following character sets: United States (code page 437), Multilingual (code page 850), Portuguese (code page 860), French-Canadian (code page 863), and Nordic (code page 865). The default character set is United States.

Although 1-2-3 stores characters from the character sets listed above, it displays and prints characters from the ANSI character set, which is the character set that Windows supports.

Changing character sets

Some drivers let you use more than one character set. If a table was created with a character set other than the character set 1-2-3 is using, you may have to use /Data External Other Translation from the 1-2-3 Classic menu to change the character set in order to read all of the data from the table. For more information about using the 1-2-3 Classic menu, see Chapter 5 in the *User's Guide*.

To change a character set

1. If you have not already done so, use Tools Database Connect to External to connect to an external table in the database whose character set you want to change.
2. Choose /Data External Other Translation from the 1-2-3 Classic menu.

   1-2-3 prompts you to enter the name of the database whose character set you want to change and displays the names of DataLens drivers.
3. Enter the name of the driver associated with the database whose character set you want to change.

   1-2-3 displays the names of databases.
4. Enter the name of the database for which you want to select a different character set.
5. Select the character set you want to use.

The character set you select remains in use for all of the external tables in the external database you specified until you end the current work session or select another character set for that external database.

Using the DataLens Driver for SQL Server Tables

The DataLens Driver for SQL Server (the SQL Server driver) is compatible with Microsoft® SQL Server 4.2. This topic describes the capabilities of the SQL Server driver and provides information you should know when you perform any operation on external SQL Server tables from within 1-2-3.

To access a Sybase SQL Server you must have the Sybase Net Library. For more information, see For Administrators.

Using the sample file

1-2-3 provides a sample worksheet file with each DataLens driver. You use the file to create an external table. See Using a 1-2-3 model range for more information. You can then use the external table with the procedures in Chapter 21 in the *User's Guide*.

The sample file for the SQL Server driver, EMPLOYEE.WK4, is in the subdirectory SAMPLE\dbase in your 1-2-3 directory. If you are using the network version of 1-2-3, SAMPLE\dbase is a subdirectory in your personal directory.

Capabilities of the SQL driver

By using the SQL Server driver with 1-2-3, you can access, modify, create, and delete existing external tables from within 1-2-3. The SQL Server driver communicates directly with SQL Server and SQL Server does much of the query work. To use the 1-2-3 Tools Database commands, Query commands, and the database @functions with external tables, you specify the range name you assigned to the external table as the database table or input range.

Basic database tasks

The table below lists the basic database tasks you can perform on external tables using the SQL Server driver and the 1-2-3 commands you use to accomplish those tasks after you connect to an external table.

Database task	Description	1-2-3 commands
Create tables	Create external tables	Tools Database Create Table
Delete rows	Delete records from external tables	Tools Database Delete Records
Drop tables	Delete external tables	/Data External Delete
Extract data	Copy records or specific fields from an external table to 1-2-3	Tools Database New Query
Insert rows	Add records to external tables	Tools Database Append Records
Update rows	Edit records in external tables	Query Update Database Table

In addition, you can use database @functions to perform calculations in 1-2-3 with data that resides in external tables. You can also use Tools Database Send Command to send commands to the SQL Server driver and to send executable SQL commands to SQL Server.

Help Help includes syntax, arguments, notes and examples for every 1-2-3 @function. For more information about 1-2-3 database @functions, search on "Database @functions" in main Help for 1-2-3.

Operators and functions

The table below contains the data analysis operators and functions 1-2-3 and the SQL Server driver send to the database for analyzing data in external tables. To use these, place them in 1-2-3 criteria.

Type	*Description*	*Example of use in 1-2-3*
Arithmetic Operators	Add	+ (plus for addition)
	Subtract	- (minus for subtraction)
	Multiply	* (asterisk for multiplication)
	Divide	/ (slash for division)
	Exponentiation	^ (caret to raise a number to a power)
	Negation	- (minus for a negative expression)
Logical Operators	Equal to	=
	Not equal to	<>
	Greater than	>
	Less than	<
	Greater than or equal to	>=
	Less than or equal to	<=
	Logical AND	Two or more criteria in a query linked by AND, two or more criteria in the same row in the criteria range, or a formula criterion that includes #AND#
	Logical NOT	Label criteria preceded by ~ (tilde) or a formula criterion preceded by #NOT#
	Logical OR	Two or more criteria in a query linked by OR, criteria in two or more rows in the criteria range, or a formula criterion that includes #OR#
	Wildcard matches	Criteria entries that use * (asterisk) or ? (question mark) to match characters (When using the SQL Server driver, you can use the asterisk wildcard character anywhere in a label criterion, not just at the end.)
Date functions	Create a date number	@DATE
	Determine the day of the month	@DAY
	Determine the hour of the day	@HOUR
	Determine the minute of the hour	@MINUTE
	Determine the	@MONTH

	month of the year	
	Determine the current date and time	@NOW
	Determine the second of the minute	@SECOND
	Create a time number	@TIME
	Determine today’s date	@TODAY
	Determine the year	@YEAR
Mathematical functions	Absolute value	@ABS
	Arc cosine	@ACOS
	Arc sine	@ASIN
	Two-quadrant arc tangent of a value	@ATAN
	Four-quadrant arc tangent of two values	@ATAN2
	Cosine	@COS
	e (the constant used as the base in natural logarithms) raised to a power	@EXP
	Integer part of a value	@INT
	Natural logarithm	@LN
	Base 10 logarithm	@LOG
	Modulus (remainder) of a division	@MOD
	Value of π	@PI
	Rounded value	@ROUND
	Sine	@SIN
	Square root	@SQRT
	Tangent	@TAN
Text functions	Concatenate strings	Using a 1-2-3 text formula, such as +“Data”&“base”, to combine character strings
	Generate a character from its ASCII value	@CHAR
	Generate the ASCII value of a character	@CODE

	Return the first characters of a string	@LEFT
	Return the length of a string	@LENGTH
	Convert a string to lowercase	@LOWER
	Repeat a string more than once	@REPEAT
	Return the last characters of a string	@RIGHT
	Convert a string to uppercase	@UPPER
	Convert a string to a value	@VALUE
Other functions	Send SQL Server operators from 1-2-3 to SQL Server	@DQUERY

Note If you use other 1-2-3 functions to analyze data in external tables, you may notice an increase in the time the analysis takes because 1-2-3 must bring all of the rows in the external table into memory in order to analyze the data.

Help Help includes syntax, arguments, notes and examples for every 1-2-3 @function. For more information about 1-2-3 database @functions, search on “Database @functions” in main Help for 1-2-3.

Creating external tables

If authorized by your database administrator, you can create external tables from within 1-2-3. To create an external table, you can use an existing external table or a 1-2-3 database table as a model and then use Tools Database Create Table, or you can create a table definition and use /Data External Create from the 1-2-3 Classic menu.

This section provides information about using a model, and describes the table definition for SQL Server tables.

Using a 1-2-3 model range

If you use a model range to create a table, the driver uses the default data types when creating the table. The following table shows 1-2-3 data types and the default data types the driver converts them to when it creates an SQL Server table.

1-2-3 data type	Data type in SQL Server table
Label	Varchar
Value	Float
Value formatted as a date or time	Datetime

To specify non-default data types, you can create a table definition and use /Data External Create from the 1-2-3 Classic menu to create an external table.

About the table definition

The table definition contains information that tells the driver how to set up the external table. A table definition contains six columns of information, although the SQL Server driver only uses information from columns 1, 2, 3, and 6, which are described below.

- Column 1 contains field names. This information is required for each field in the table.

  When using the SQL Server driver, the maximum number of fields in an external table is 250.

  Field names can contain as many as 30 characters and can contain letters, numbers, _ (underscores), and $ (dollar signs). The first character of a field name must be a letter or underscore.

  Field names within a table must be unique.

  You cannot use SQL reserved words as field names. For a list of these reserved words, see the SQL Server Language Reference.
- Column 2 contains data types. This information is required for each field in the table. For information about data types, see “About SQL Server data types” below.
- Column 3 contains field widths. Many data types have a fixed field width and ignore the field width you enter in the table definition. For information about the field widths allowed with each data type, see “About SQL Server data types” below.
- Column 6 contains field creation strings. You use field creation strings with the SQL Server driver to specify whether a field accepts null entries. In 1-2-3, a blank cell and @NA in a cell are null entries. If you do not want a field to accept null entries, use the creation string NOT NULL for that field in the table definition. If you want a field to accept null entries, do not enter any creation string. Fields with binary, bit, or char data types are always created as NOT NULL, regardless of the creation string you enter.

The figure below shows a table definition for an SQL Server table. Although columns 4 and 5 must be present in the table definition, the SQL Server driver ignores entries in these columns because they do not apply to SQL Server tables.

Field names	**Data types**	**Field widths**	**Column labels**	**Field descriptions**	**Field creation strings**
EMPID	float	8	NA	NA	NA
LAST	varchar	12	NA	NA	NA
DOH	datetime	8	NA	NA	NA
SALARIED	float	8	NA	NA	NA
DEPTNUM	float	8	NA	NA	NA
TIMESTAMP	timestamp	8	NA	NA	NA

About SQL Server data types

Databases let you use many different types of data. It is important for you to know the data types SQL Server supports when you create new tables (as well as when you enter and analyze data). 1-2-3 uses these data types in table definitions when you use /Data External Create.

When using the SQL Server driver, you can send SQL commands directly to the SQL Server. This is useful for submitting administrative commands and using SQL Server stored procedures. However, you cannot send commands directly to SQL Server that would result in SQL Server returning data to 1-2-3. For example, you cannot use SELECT or stored procedures that return data. To use an SQL command to return data from SQL Server, extract data into an SQL Server view, and perform a query on that view.

When using SQL Server, you can also send driver and database configuration options to the driver to override current settings for some of the options available in the DC parameter in the registration file. The configuration options you can override are $ATb, $CPn, $SDn, $TSb and $OUt. For an explanation of these options, see Database configuration options or your database administrator.

When you send data between 1-2-3 and SQL Server, the SQL Server driver converts SQL Server data types to 1-2-3 data types, and vice versa, so both programs can use and understand the same data. The table below shows SQL Server data types and the data types the driver converts them to in 1-2-3. The table also shows the field widths (the storage sizes in SQL Server databases) and the values allowed for the various data types.

SQL Server data type	**Field width**	**Values accepted by SQL Server**	**Data type in 1-2-3**
Binary	1-255	Binary data equal to the length specified by the field width	None
Bit	1	0, 1	Value
Char	1-255	Character strings equal to the length specified by the field width	Label
Datetime	8	Dates from January 1, 1753 through December 31, 9999 and all times of day	Value (date or time number)
Float	8	±1.7E-308 through ±1.7E+308 with 15-digit precision	Value
Image	16	Binary data up to 2,147,483,647 bytes	None
Int	4	-2,147,483,647 through 2,147,483,647	Value
Money	8	-922,337,203,685,447.5807 through 922,337,203,685,447.5807	Value
Real	4	~3.4E-38 through 3.4E+38	~
Small-datetime	4	January 1, 1900 through June 6, 2079 and all minutes	Value (date or time number)
Smallint	2	-32,767 through 32,767	Value
Smallmoney	4	+214,748.3647 through -214,748.3648	Value
Text	16	Character strings up to 2,147,483,647 characters in length	Label
Timestamp	8	Binary row ID	None
Tinyint	1	0 through 255	Value
Varbinary	1-255	Binary data up to the length specified by the field width	None
Varchar	1-255	Character strings up to the length specified by the field width	Label

Notes about SQL Server data types

- SQL Server uses the field width in the table definition only when creating fields that have char, varchar, binary, or varbinary as the data type. SQL Server ignores the field width for all other data types.
- Do not include in a 1-2-3 query table or criteria range the name of a field whose SQL Server data type does not have a 1-2-3 data type (SQL Server data types whose 1-2-3 data type is None in the table above). The data in these fields is usable by SQL Server only. If you extract these data types into 1-2-3, 1-2-3 displays ERR. You can, however, include these data types in the table definition when you create a table.
- If you enter more characters than the field width allows, SQL Server truncates the entry after the last character the field width allows.
- If the SQL Server driver is set to automatically add a timestamp field to new tables, which is the default setting, do not include a timestamp field in the table definition that you use to create a table. If you are not sure if the driver is set to add a timestamp field, see your database administrator.
- Dates from SQL Server are entered as date numbers in 1-2-3 and formatted in the date format specified in Tools User Setup International. Times from SQL Server are entered as time numbers in 1-2-3 and formatted in the time format specified in Tools User Setup International. Datetimes from SQL Server are entered as datetime numbers (values with a date number preceding the decimal point and a time number following the decimal point) in 1-2-3 and formatted in the date format specified in Tools User Setup International.

Although 1-2-3 can only display dates from January 1, 1900 through December 31, 2099, it can store all dates from SQL Server and return them to SQL Server. When you format cells that contain dates outside the range that 1-2-3 can display, 1-2-3 displays *** (asterisks) in the cells.

- Although 1-2-3 can only display numbers from ±1.0E-99 through ±9.99E+99, it can store all numbers from SQL Server tables and return them to the tables. 1-2-3 displays *** (asterisks) in cells that contain numbers outside the range that 1-2-3 can display.
- Binary, bit, and char SQL Server fields are always created NOT NULL, regardless of the creation string you enter in column 6 of the table definition when you create the table. Fields created NOT NULL never allow null entries.
- Datetime, float, image, int, money, real, smalldatetime, smallint, smallmoney, text, tinyint, varchar, and varbinary SQL Server fields allow null entries unless you enter NOT NULL as the creation string in column 6 of the table definition when you create the table.
- Null entries from SQL Server become blank cells in 1-2-3.
- You cannot create tables that contain user-defined data types.
- If you use a table whose name exceeds 15 characters, 1-2-3 displays ???? (question marks) for the range name of that table.
- If you are using the SQL Server driver, you can use @DQUERY in 1-2-3 to access SQL Server built-in functions, such as DATALENGTH, which returns the length of variable length data. For example, if you are connected to the table STORES in the database PUBS and you want to extract all store names that have fewer than 30 characters, you would enter the following in the criteria range: @DQUERY(“DATALENGTH”,STOR_NAME)<30
- If you are using the SQL Server driver to query a table that does not contain a timestamp field, the query table may contain duplicate records. If you subsequently attempt to copy the modified records back to the external table, you may not be able to do so. In this case, you should qualify the criteria to limit duplicate records.
- If you are using the SQL Server driver and want to use a table created by someone else, you must have authorization and may have to enter an owner name (usually the user ID of the person who created the table) as part of the table name. If required, type the owner name, a space, and the table name.
- When you use 1-2-3 and the SQL Server driver to query a case-sensitive server, use aggregate formulas only when field names in the external table are stored in uppercase. If the field names are not stored with uppercase characters and you want to generate a summary table, create a 1-variable table using Range Analyze What-if Table.

Using macros for transaction control

When you use a macro to modify an external table, the database management system does not actually modify the table until the driver commits (finalizes) the modification transaction. The driver commits a transaction when any of the following occurs:

- The macro containing a command that modifies the external table ends.
- A command that modifies the external table through the menus is executed.
- The macro uses the advanced macro command {COMMIT}.

When a command is waiting to be committed, no one using a different program can access the table that the command modifies. If the macro uses Tools Database Disconnect or the connection to the external table disconnects in some other way, the driver does not commit the command that is waiting but instead rolls back (deletes) the command.

To be sure macro transactions are committed, use {COMMIT} after each of the following commands, which insert and delete records, and update tables: Tools Database Append Records, Tools Database Delete Records, and Query Update Database Table.

{COMMIT} uses two optional arguments: {COMMIT *“drivername”, “databasename”*}. If you use the optional arguments, the driver commits only the transaction pending for the driver and database you specify. Be sure to enclose the driver name and database name in “ ” (quotation marks). If you do not use the optional arguments, the driver commits all pending transactions.

Use the advanced macro command {ROLLBACK} to cancel a command in a macro that modifies an external table. To be effective, you must use {ROLLBACK} before the macro commits the command you want to roll back. {ROLLBACK} also uses two optional arguments: {ROLLBACK *“drivername”, “databasename”*}.

Supported character sets

The SQL Server driver supports the following character sets: United States (code page 437), Multilingual (code page 850), Portuguese (code page 860), French-Canadian (code page 863), and Nordic (code page 865). The default character set is United States.

Although 1-2-3 stores characters from the character sets listed above, it displays and prints characters from the ANSI character set, which is the character set that Windows supports.

Changing character sets

Some drivers let you use more than one character set. If a table was created with a character set other than the character set 1-2-3 is using, you may have to use /Data External Other Translation from the 1-2-3 Classic menu to change the character set in order to read all of the data from the table. For more information about using the 1-2-3 Classic menu, see Chapter 5 in the *User’s Guide*.

To change a character set

1. If you have not already done so, use Tools Database Connect to External to connect to an external table in the database whose character set you want to change.
2. Choose /Data External Other Translation from the 1-2-3 Classic menu.

   1-2-3 prompts you to enter the name of the database whose character set you want to change and displays the names of DataLens drivers.
3. Enter the name of the driver associated with the database whose character set you want to change.

   1-2-3 displays the names of databases.
4. Enter the name of the database for which you want to select a different character set.
5. Select the character set you want to use.

The character set you select remains in use for all of the external tables in the external database you specified until you end the current work session or select another character set for that external database.

For Administrators using dBase IV

Among the files that the 1-2-3 Install program transfers to the 1-2-3 directory is the registration file, which is named LOTUS.BCF. The purpose of the registration file is to give 1-2-3 information about the DataLens drivers and databases you want to make available to your users. 1-2-3 uses the information in the registration files to browse and connect to drivers and databases.

The registration file consists of driver records and database records.

Driver records tell 1-2-3 which drivers to list when users choose a command that displays driver names, such as Tools Database Connect to External and which parameters to use with those drivers. The registration file must include a driver record for each DataLens driver you want to make available to your users.

Database records tell 1-2-3 which databases to list when users select Tools Database Connect to External. These records are optional.

This section describes the following:

- Driver records and the parameters you can include in a driver record
- The syntax of a driver record and a database record
- Database records and the parameters you can include in a database record

Driver records

When users install the dBASE IV driver, the Install program automatically adds a driver record to the registration file to give users access to the dBASE IV driver. The driver record looks like this:

```
DN="dBASE_IV" DL="DLODBC"
DD="DataLens Driver for dBASE IV Tables, Release 2.0"
DC="driver=LODBF04.DLL";
```

As the database administrator, you may want to change the parameters in a driver record in order to customize the driver for your users. You may also want to add records to the registration file in order to make additional drivers available to your users.

Driver record parameters

The registration file must contain at least one driver record. A driver record contains several parts, called parameters. This section shows the parameters you can use in a driver record.

The table below describes the parameters you can use when creating or editing a driver record for the dBASE IV driver. Two of the parameters are required.

Parameter	Required	Example	Description
DN=“*Drivername*”	Yes	DN=“dBASE_IV”	DN identifies the record as a driver record and must be the first parameter in the record. *Drivername* is the name you want 1-2-3 to display to the user. The driver name cannot include spaces, must be enclosed in “ ” (quotation marks), and includes up to 80 characters. This name

			must be unique for each driver record.
DL="*DriverFilename*"	Yes	DL="DLODBC"	DL specifies the file name (without the extension) of the driver program. (DLODBC for the dBASE IV driver). Enclose the name in “ ” (quotation marks). 1-2-3 requires the file to be in the 1-2-3 directory.
DD="*Driver Description*"	No	DD="DataLens Driver for dBASE IV Tables, Release 2.0"	DD specifies a description of the driver. Use this parameter as annotation in the registration file. 1-2-3 displays this description to the user when the user selects a driver. Enclose the description in “ ” (quotation marks).
DC="*Driver Configuration Information*"	No	DC="Driver= LODBF04.DLL"	DC sends configuration information to the driver. Enclose the configuration information in “ ” (quotation marks).

Driver and database record syntax

Follow the rules below when adding or editing records in the registration file.

Parameter syntax

- Use = (equal sign) to separate the parameter name from the value of the parameter, such as DN="dBASE_IV". Do not include a space before or after the equal sign.
- When using the DN, DL, and DD parameters, enclose the value of the parameter in " " (quotation marks), such as DL="DLODBC".

Record syntax

- Include at least one space between parameters in a record.
- Include a ; (semicolon) after the last parameter in record. 1-2-3 interprets all parameters as being part of one record until it encounters a semicolon. A record can wrap to several lines.
- Enter driver and database records in the registration file in the order you want 1-2-3 to list driver and database names when it displays them.

Database records

Database records are optional. Database records tell 1-2-3 to list specific databases (directories) when a user chooses a command that displays database names, such as Tools Database Connect to External. If there are no database records, 1-2-3 displays the current directory only.

> **Note** When 1-2-3 displays the names of databases, it first displays the names of registered databases (databases that have a database record in the registration file) and then displays the name of the current directory.

Database record parameters

The table below describes the parameters you can use when creating or editing a database record for the dBASE IV driver. Two of the parameters are required.

Parameter	Required	Example	Description
DB="*Databasename*"	Yes	DB="C:\personnnel"	When using the dBASE IV driver, database is the directory in which dBASE files reside. DB identifies the record as a database record and must be the first parameter in the

			record. *Databasename* specifies the database (path) name. Do not include spaces in the database name. The database name includes up to 80 characters. Enclose the name in " " (quotation marks).
DN=“*Drivername*”	Yes	DN="dBASE_IV”	DN specifies the name of the driver as you specified it in the DN parameter of the driver record. In a database record, the DN parameter must follow the DB parameter. Enclose the name in " " (quotation marks). If the driver name in the database record does not match the driver name in any driver record in the registration file, 1-2-3 ignores this database record.
DD=“*Database Description*”	No	DD=“Employees throughout the world”	DD specifies a description of the database. Use this parameter as annotation in the registration file. The database description can include up to 80 characters. Enclose the description in “ ” (quotation marks).

A database record may look like the following:

```
DB="c:\personnel" DN="dBASE_IV"
DD="Employees throughout the world";
```

Database locking

When sharing files on a network, by default 1-2-3 writes to a table only if no one else is using the table. However, multiple users can read the table concurrently.

If your users are operating exclusively as 1-2-3 users, you can override the default setting of Locking=File and activate the Locking=Record setting. Locking at the record level means that multiple users can update the same network file concurrently and database integrity is maintained.

To set database locking options

1. In the Main Group of the Program Manager Window, double-click the Control Panel icon.
2. In the Control Panel window, double-click the ODBC icon.
3. In the Data Sources dialog box, choose Add.
4. In the Add Data Sources dialog box, select "Lotus Q+E dBASE" from the list of installed drivers, and choose OK.
5. Enter the options you want in the ODBC dBASE Driver Setup dialog box and choose OK.

The options that you enter in this dialog box will also appear in the ODBC.INI file. Also, the Data Source Name that you enter will appear in the 1-2-3 Connect to External dialog box.

Database and driver communication and security

Keep in mind the following points concerning security and communication as you work with the dBASE IV driver:

- The dBASE IV driver works directly with dBASE IV tables. Users do no have to use dBASE IV to access tables.
- When a user deletes a table, the dBASE IV driver deletes .MDX files named the same as the table being deleted, but does not delete associated forms, reports, or index files.

Most write operations in 1-2-3 are accomplished by using Tools Database Append Records, Tools Database Delete Records, and Query Update Database Table. Read operations are accomplished by using Tools Database New Query.

- If a dBASE IV table is read-only, users can read the data in the table but cannot write to the table, delete records in the table, or delete the table.
- You can use standard operating system commands or network security mechanisms to control user access to dBASE IV tables.

Creating standard registration files

If you are an administrator using the server edition of 1-2-3, you can create a standard registration file and store it on the server in the default installation directory. When users rerun Node installation, the standard registration file is copied to the appropriate directory.

The sequence of records in the registration file determines the order in which drivers and databases are listed for the user. To maintain a consistent user interface and support shared macros, you can create and distribute one registration file for all users or each group of users.

For Administrators Using Paradox

Among the files that the 1-2-3 Install program transfers to the \Lotusapp\Datalens directory is the registration file, which is named LOTUS.BCF. The purpose of the registration file is to give 1-2-3 information about the DataLens drivers and databases you want to make available to your users. 1-2-3 uses the information in the registration file to browse and to connect to drivers and databases.

The registration file consists of driver records and database records.

Driver records tell 1-2-3 which drivers to list when users choose a command that displays driver names, such as Tools Database Connect to External, and which parameters to use with those drivers. The registration file must include a driver record for each DataLens driver you want to make available to your users.

Database records tell 1-2-3 which databases to list when users choose Tools Database Connect to External. Database records are optional.

This section describes the following:

- Driver records and the parameters you can include in a driver record
- -The syntax of a driver record and a database record
- Database records and the parameters you can include in a database record

Driver records

When users use the 1-2-3 Install program to install the Paradox driver, the Install program automatically adds a driver record to the registration file to give users access to the Paradox driver. The driver record looks like this:

DN=“Paradox” DL=“PARALENW"

DD=“DataLens Driver for Paradox Tables” AC=UI,PW

DC=“*location of the network control file*”;

> **Note** If your Paradox files are not on a network, the Install program does not include the parameter DC=“*location of the network control file*” in the registration file.

As the database administrator, you may want to change the parameters in a driver record in order to customize the driver for your users. You also may want to add records to the registration file in order to make additional drivers available to your users.

Driver record parameters

The registration file must contain at least one driver record. A driver record contains several parts, called parameters. This section shows the parameters you can use in a driver record.

The table below describes the parameters you can use when creating or editing a driver record for the Paradox driver. Two of the parameters are required (three if your Paradox files are on a network).

Parameter	Required	Example	Description
DN=“*Drivername*”	Yes	DN=“Paradox”	DN identifies the record as a driver record and must be the first parameter in the record. *Drivername* is the name you want 1-2-3 to display

			to the user. The driver name cannot include spaces, must be enclosed in “ ” (quotation marks), and includes up to 80 characters. This name must be unique for each driver record.
DL=“*DriverFilename*”	Yes	DL=“PARALENW”	DL specifies the file name (without the extension) of the driver program (PARALENW for the Paradox driver). Enclose the name in “ ” (quotation marks). 1-2-3 requires the file to be in the 1-2-3 directory.
DD=“*Driver Description*”	No	DD=“DataLens Driver for Paradox Tables”	DD specifies a description of the driver. Use this parameter as annotation in the registration file. The driver description includes up to 80 characters. Enclose the description in “ ” (quotation marks).
AC=UI,PW	No	AC=UI,PW	AC is the access control parameter. AC=UI causes 1-2-3 to prompt for a user ID when a user connects to a driver. The user ID identifies who is using a table when someone is denied access to the table. If a user does not enter a user ID when prompted, the driver uses the default ID, ParaLens. If you include PW in this parameter (AC=UI,PW), 1-2-3 also prompts for a password when a user connects to a driver.
DC=“*Driver Configuration Information*”	If Paradox files are on a network	DC=“P:\ PDOXDATA” or DC=“NOSHARE”	DC lists the path to the network control file, PARADOX.NET. If PARADOX.NET does not exist in the specified location, the driver creates PARADOX.NET. This parameter is required only if users will be accessing shared tables on a network drive. Enclose the path in “ ” (quotation marks). If you want to give a user access to files on the network without using the network control program, specify “NOSHARE” as the path.

Driver and database record syntax

This section describes the syntax rules for records in the registration file. Follow the rules below when adding or editing records in the registration file.

Parameter syntax

- Use = (equal sign) to separate the parameter name from the value of the parameter, such as DN=“Paradox”. Do not include a space before or after the = (equal sign).
- When using the DN, DL, DC, and DD parameters, enclose the value of the parameter in “ ” (quotation marks), such as DL=“PARALENW”.

Record syntax

- Include at least one space between parameters in a record.
- Include a ; (semicolon) after the last parameter in a record. 1-2-3 interprets all parameters as being part of one record until it encounters a semicolon. A record can wrap to several lines.
- Enter driver and database records in the registration file in the order you want 1-2-3 to list driver and database names when it displays them.

Database records

Database records are optional. Database records tell 1-2-3 to list specific databases (directories) when a user chooses a command that displays database names, such as Tools Database Connect to External. If there are no database records, 1-2-3 displays the current directory only.

> **Note** When 1-2-3 displays the names of databases, it first displays the names of registered databases (databases that have a database record in the registration file) and then displays the name of the current directory. If the registration file contains a database record for the current directory, 1-2-3 displays the name of that directory twice.

Database record parameters

The table below describes the parameters you can use when creating or editing a database record for the Paradox driver. Two of the parameters are required.

Parameter	Required	Example	Description
DB="*Databasename*"	Yes	DB="c:\personnel"	DB identifies the record as a database record and must be the first parameter in the record. *Databasename* specifies the database (path) name. Do not include spaces in the database name. Enclose the name in " " (quotation marks).
DN="*Drivername*"	Yes	DN="Paradox"	DN specifies the name of the driver as you specified it in the DN parameter of the driver record. In a database record, the DN parameter must follow the DB parameter. Enclose the name in " " (quotation marks). If the driver name in the database record does not match the driver name in any driver record in the registration file, 1-2-3 ignores this database record.
DD="*Database Description*"	No	DD="Employees throughout the world"	DD specifies a description of the database. Use the DD parameter as annotation in the registration file. The database description includes up to 80 characters. Enclose the description in " " (quotation marks).
AC=UI,PW	No	AC=UI,PW	AC is the access control parameter. AC=UI causes 1-2-3 to prompt for a user ID when a user connects to a driver. The user ID identifies who is using a table when someone is denied access to the table. If a user does not enter a user ID when prompted, the driver uses the default

			ID, ParaLens. If you include PW in this parameter (AC=UI,PW), 1-2-3 also prompts for a password.

DC="*Driver Configuration Information*"	No	DC="P:\ PDOXDATA"
		or
		DC="NOSHARE"
			DC lists the path to the network control file, PARADOX.NET. Use this in the database record only if this database is on a network that is completely independent from a network control file that is already specified in the DC parameter of the driver record. If you want to give a user access to files on the network without using the network control program, specify "NOSHARE" as the path in the DC parameter. You can also specify "NOSHARE" as the path to give the user access to a specific database on a local drive while also sharing other files on the network. Enclose the path in " " (quotation marks).

A database record may look like the following:

```
DB="c:\personnel" DN="Paradox"
DD="Employees throughout the world" AC=UI;
```

Database and driver communication and security

Keep in mind the following points concerning security and communication as you work with the Paradox driver:

- The Paradox driver works directly with Paradox tables. Users do not have to use Paradox to access the tables.
- When a user deletes a table, the Paradox driver also deletes the associated primary index (.PX) file, if it exists. The driver does not delete other associated files, such as forms, reports, scripts, or secondary index files.
- On a network, you can use 1-2-3 to write to a table only if no one else is using the table. However, multiple users can read a table concurrently. In addition, you can use 1-2-3 to read a table when someone is using Paradox to co-edit the table.

  Note Most write operations in 1-2-3 are accomplished by using Tools Database Append Records, Tools Database Delete Records, and Query Update Database Table. Read operations are accomplished by using Tools Database New Query.
- To prevent users from accessing a Paradox table, encrypt the file from within Paradox or use Tools Database Send Command and send the ENCRYPT command to the driver. For more information about the ENCRYPT command, see Table SecurityF.
- To let users read a table but prevent them from modifying or deleting the table, write-protect the table from within Paradox.

Creating standard registration files

If you are an administrator using the server edition of 1-2-3, you can create a standard registration file and store it on the server in the default installation directory. When users rerun Node installation, the standard registration file is copied to the appropriate directory.

The sequence of records in the registration file determines the order in which drivers and databases are listed for the user. To maintain a consistent user interface and support shared macros, you can create and distribute one registration file for all users or each group of users.

Note The DC parameter in the driver record in the registration file lists the directory that contains the network control program. To use the same registration file, all network users must connect to the network using the same drive letter that is specified in the DC parameter. (You specify this letter when you first install 1-2-3 on the network.)

For Administrators Using SQL Server

Among the files that the 1-2-3 Install program transfers to the \Lotusapp\Datalens directory is the registration file, which is named LOTUS.BCF. The purpose of the registration file is to give 1-2-3 information about the DataLens drivers and databases you want to make available to your users. 1-2-3 uses the information in the registration file to browse and to connect to drivers and databases.

The registration file consists of driver records and database records.

Driver records tell 1-2-3 which drivers to list when users choose a command that displays driver names, such as Tools Database Connect to External, and which parameters to use with those drivers. The registration file must include a driver record for each DataLens driver you want to make available to your users.

Database records tell 1-2-3 which databases to list when users choose Tools Database Connect to External. Database records are optional.

This section describes the following:

- Driver records and the parameters and configuration options you can include in a driver record
- The syntax of a driver record and a database record
- Database records and the parameters and configuration options you can include in a database record

Driver records

When users use the 1-2-3 Install program to install the SQL Server driver, the Install program automatically adds a driver record to the registration file to give users access to SQL Server. The driver record looks like this:

DN="SQLserver_*servername*" DL="DLSMW"
DD="DataLens Driver for SQL Server" AC=UI,PW DC="$CF SQLSDLW $SV LVASQLSW $OP *servername*";

As the database administrator, you may want to change the parameters or options in a driver record in order to customize the driver for your users. You also may want to add records to the registration file in order to make additional drivers available to your users.

Driver record parameters

The registration file must contain at least one driver record. A driver record contains several parts, called parameters. This section shows the parameters you can use in a driver record.

The table below describes the parameters you can use when creating or editing a driver record for the SQL Server driver. Four of the parameters are required.

Parameter	Required	Example	Description
DN="*Drivername*"	Yes	DN="SQLserver_ YOURSERVER"	DN identifies the record as a driver record and must be the first parameter in the record. *Drivername* is the name you want 1-2-3 to display to the user. The driver name cannot include

			spaces, must be enclosed in “ ” (quotation marks), and can be up to 80 characters long. This name must be unique for each driver record.
DL=“*DriverFilename*”	Yes	DL=“DLSMW”	DL specifies the file name (without the extension) of the driver program (DLSMW). Enclose the name in “ ” (quotation marks). The file must be in the 1-2-3 directory.
DD=“*Driver Description*”	No	DD=“DataLens Driver for SQL Server”	DD specifies a description of the driver. Use this parameter as annotation in the registration file. The driver description includes up to 80 characters. Enclose the description in “ ” (quotation marks).
AC=UI,PW	Yes	AC=UI,PW	The access control parameter, AC=UI,PW, specifies that the user must enter a user ID and a password.
DC=“*Driver Configuration Information*”	Yes	DC=“$CF SQLSDLW $SV LVASQLSW $OP YOURSERVER”	DC sends configuration information to the driver. Enclose the configuration information in “ ” (quotation marks). For a list of the options you can include in the DC parameter, see the table in “Driver configuration options,” which follows.

Driver configuration options

The table below lists the options you can include as part of the driver configuration (DC) parameter.

> **Note** The table indicates several types of arguments these options can use. The argument types are *b* (boolean — 1 or 0),*n* (number), *t* (text), and *f* (file name).

Option	**Required**	**Example (including delimiter)**	**Description**
AT*b*	No	$AT1	The AT (all tables) option controls which tables 1-2-3 displays when users choose a command that prompts for a user name and then lists the tables in a database, such as Tools Database Connect to External. If AT is set to the default (AT1), 1-2-3 displays all tables in the database. If AT is set to 0 (AT0), 1-2-3 displays only tables created with the user’s owner name.
CF*f*	Yes	$CF SQLSDLW	CF (configuration file) specifies the name of the driver configuration file, which contains information about the character set the driver uses, as well as the names and locations of the message and template files. The message file contains messages, such as error messages, status messages, and application-specific options. The template file supplies the syntax SQL Server uses in formatting SQL strings.
CP*n*	No	$CP437	CP (code page) specifies the code page of the data stored on the server. The default is 437 (Code Page 437).

OP*t*	Yes	$OP YOURSERVER	OP (open path) specifies the name of the file server on which SQL Server resides.
SB*n*	No	$SB3072	SB (statement buffer) specifies the size of the SQL statement buffer in bytes. The size of the buffer determines the maximum size of the SQL statement the driver can send to the server. The default is 3072, which is the maximum size. The minimum is 255.
SD*n*	No	$SD15	SD (significant digits) specifies the number of significant digits the SQL Server driver uses for conversion from 8-byte real numbers to strings. The default is 15. The other acceptable values are 14 and 16. The SD value can affect whether criteria in a 1-2-3 criteria range match real numbers in an external table.
SV*f*	Yes	$SV LVASQLSW	SV specifies the name of the system services library module.
TS*b*	No	$TS1	TS (timestamp) specifies whether to generate a timestamp field automatically when creating an external table. TS1 generates a timestamp field; TS0 does not. The default is TS1. Updates may not work on tables that do not contain timestamp fields because the driver may not be able to uniquely identify rows.

Driver and database record syntax

This section describes the syntax rules for records in the registration file and gives an example of a driver record for the SQL Server driver.

When adding or editing records or parts of records in the registration file, follow the rules below.

Parameter syntax

- Use = (equal sign) to separate the parameter name from the value of the parameter, such as DN="SQLserver_YOURSERVER".
- When using the DN, DL, DC, and DD parameters, enclose the value of the parameter in " " (quotation marks), such as DL="DLSMW".

Configuration option syntax

- Precede each option in the DC parameter with a delimiter. The delimiter can be any character that does not appear in the command itself. The driver automatically considers the first character of any option statement to be the delimiter.
- Space between an option name and an option value is optional. For example, AT1 is the same as AT 1.

Record syntax

- Include at least one space between parameters in a record.
- Include a ; (semicolon) after the last parameter in a record. 1-2-3 interprets all parameters as being part of one record until it encounters semicolon. A record can wrap to several lines.
- Enter driver and database records in the registration file in the order you want 1-2-3 to list driver and database names when it displays them.

Example of driver record

The following example shows the default driver record for the SQL Server driver. This driver record is created by the 1-2-3 Install program after it prompts the user for the name of the server on which SQL Server is installed. In this record, the name of the server is YOURSERVER. The delimiter in this record is $ (dollar sign), although you can use any delimiter you want. The tables that follow the example explain each parameter and configuration option in the record.

```
DN=“SQLserver_YOURSERVER” DL=“DLSMW” DD=“DataLens Driver for SQL Server” AC=UI,PW
DC=“$CF SQLSDLW $SV LVASQLSW $OP YOURSERVER”;
```

The table below explains each parameter in the default driver record.

Parameter	Description
DN=“SQLserver_YOURSERVER”	Specifies “SQLserver_YOURSERVER” as the driver name. It is also the name 1-2-3 displays when users choose a command that displays driver names, such as Tools Database Connect to External.
DL=“DLSMW”	Identifies DLSMW as the name of the driver program.
DD=“DataLens Driver for SQL Server”	Annotates the driver as being the DataLens driver for SQL Server.
AC=UI,PW	Tells the driver to prompt for both a user ID and a password.
DC=“$CF SQLSDLW $SV LVASQLSW $OP YOURSERVER”	Consists of several parts, all of which are required. The parts are explained in the table below.

The table below explains each DC option in the default driver record.

DC option	Description
$CF SQLSDLW	Indicates that the name of the configuration file is SQLSDLW.
$SV LVASQLSW	Indicates that the name of the systems services file is LVASQLSW.
$OP YOURSERVER	Indicates the name of the server is YOURSERVER.

Database records

Database records are optional. Database records tell 1-2-3 to list specific databases when a user chooses a command that displays database names, such as Tools Database Connect to External. The SQL Server driver automatically browses for databases even if there are no database records in the registration file. You can add database records to the registration file to customize the options for a particular database rather than using the default options or the options you specified in the driver record.

> **Note** When 1-2-3 displays the names of databases, it first displays the names of registered databases (databases that have a database record in the registration file) and then displays the names of non-registered databases on the server.

Database record parameters

Similar to the driver record, the database record has several parameters. This section tells you the parameters you can use in a database record.

When creating or editing a database record, you can use the following parameters, two of which are required:

Parameter	Required	Example	Description
DB=“*Databasename*”	Yes	DB=“personnel”	DB identifies the record as a database record and must be the first parameter in the record. *Databasename* must be the

			same as the name of the database on the server. Do not include spaces in the database name. The database name includes up to 80 characters. Enclose the name in “ ” (quotation marks).
DN=“*Drivername*”	Yes	DN=“SQLserver_ YOURSERVER”	DN specifies the name of the driver as you specified it in the DN parameter of the driver record. In a database record, the DN parameter must follow the DB parameter. Enclose the name in “ ” (quotation marks). If the driver name in the database record does not match the driver name in any driver record in the registration file, 1-2-3 ignores this database record.
DD=“*Database Description*”	No	DD=“Employees throughout the world”	DD specifies a description of the database. Use this parameter as annotation in the registration file. The database description includes up to 80 characters. Enclose the description in “ ” (quotation marks).
DC=“*Database Configuration Information*”	No	DC=“$AT0 $OU BMARTIN $SB1000”	DC sends configuration information about this database to the driver. Enclose the configuration information in “ ” (quotation marks). For a list of the options you can include in the DC parameter, see the table in “Database configuration options,” which follows.

Database configuration options

The following table describes the options you can include as part of the database configuration (DC) parameter.

> **Note** The following table indicates several types of arguments these options can use. The argument types are *b* (boolean — 1 or 0), *n* (number), and *t* (text).

Option	Required	Example (including delimiter)	Description
AT*b*	No	$AT1	For the database specified by DB in this database record, overrides the AT setting in the driver record.
CP*n*	No	$CP437	CP (code page) specifies the code page of the data stored in the server. The default is 437 (Code Page 437).
OU*t*	No	$OU BMARTIN	OU overrides the AT option when displaying 1-2-3 tables for a database. When you enter an owner name in OU, 1-2-3 displays tables created with that owner name, as well as system tables.
SB*n*	No	$SB3072	For the database specified by DB in this database record, overrides the SB setting in the driver record.
SD*n*	No	$SD15	For the database specified by DB in this database record, overrides the SD setting in the

			driver record.
TS*b*	No	$TS1	For the database specified by DB in this database record, overrides the TS setting in the driver record.

Note Users can override many of the driver and database configuration options by using Tools Database Send Command and entering the options as driver-specific external commands. The options you can override are $AT*b*, $CP*n*, $SD*n*, $TS*b*, and $OU*t*. For more information about using Tools Database Send Command, see Chapter 22 in the *User's Guide*.

Creating standard registration files

If you are an administrator using the server edition of 1-2-3, you can create a standard registration file and store it on the server in the default installation directory. When users rerun Node installation, the standard registration file is copied to the appropriate directory.

The sequence of the records in the registration file determines the order in which drivers and databases are listed for the user. To maintain a consistent user interface and support shared macros, you can create and distribute one registration file for all users or each group of users.

SQL Server Net-Library technology

The version of DB-Library included with the SQL Server driver has the capability of separating all network-specific code into a separate program file, called a Net-Library or netlib. Netlibs let you access SQL Server on various platforms. The netlib file included with 1-2-3 provides access to OS/2 SQL Server. To obtain netlibs for other platforms, such as UNIX® or VMS, contact Sybase, Inc. at 1-800-8SYBASE.

Accessing Sybase SQL Server on other platforms

If you wish to access a Sybase SQL Server on another platform and you have installed the necessary Sybase Windows Net-Library client environment, specify the name of the WIN.INI entry at the Server Name prompt.

For example, the Sybase Net-Library installation procedure creates a SQL Server section in the Windows WIN.INI file with a default entry of DSQUERY. This entry is followed by the name of the Net-Library DLL and the necessary connection information required by that Net-Library.

```
[SQLSERVER]
DSQUERY=Net-Library DLLname, Connection information
```

You specify DSQUERY at the Server Name prompt. The installation procedure inserts a SQL Server driver record into the LOTUS.BCF. If you wish to connect to multiple servers, insert a new driver record into your LOTUS.BCF file.

For more information about configuring the Sybase Net-Library, refer to the *Sybase Installation and Reference Manual for Open Client Net-Library for Windows*.

Copyright

Lotus Development Corporation
55 Cambridge Parkway
Cambridge, Ma 02142

Your Application

As used here, **your application** refers to Lotus 1-2-3.   Applications are programs that perform a task or group of related tasks for you.

Driver

A program that lets 1-2-3 access data from a specific external data source, such as a dBASE IV table.

Data Source
Refers to the location of data you are accessing. It can be a DBMS or an external table.

Tables

The term “tables” refers to dBASE IV .DBF files.