Thoroughbred® Query-IV™
Reference Manual

Version 8.6.0

285 Davidson Ave., Suite 302 • Somerset, NJ 08873-4153
Telephone: 732-560-1377 • Outside NJ 800-524-0430
Fax: 732-560-1594

Internet address: **http://www.tbred.com**

Published by:
Thoroughbred Software International, Inc.
285 Davidson Ave., Suite 302
Somerset, New Jersey 08873-4153

Document Number: QU8.6.0M101

INTRODUCTION

Thoroughbred Query-IV is a major component of Thoroughbred IDOL-IV. It uses Thoroughbred Dictionary-IV to create queries.

Thoroughbred Dictionary-IV is a primary component of Thoroughbred IDOL-IV a fourth generation language (4GL). It is a dictionary-driven system that provides a database manager, menu manager, screen developer, message dictionary, and help manager.

This manual serves as a reference document. To find specific information, you may read the section in this Introduction titled *How this Manual is Organized*.

The other major components of IDOL-IV are Thoroughbred Script-IV and Thoroughbred Report-IV. These components require separate manuals to describe their functions.

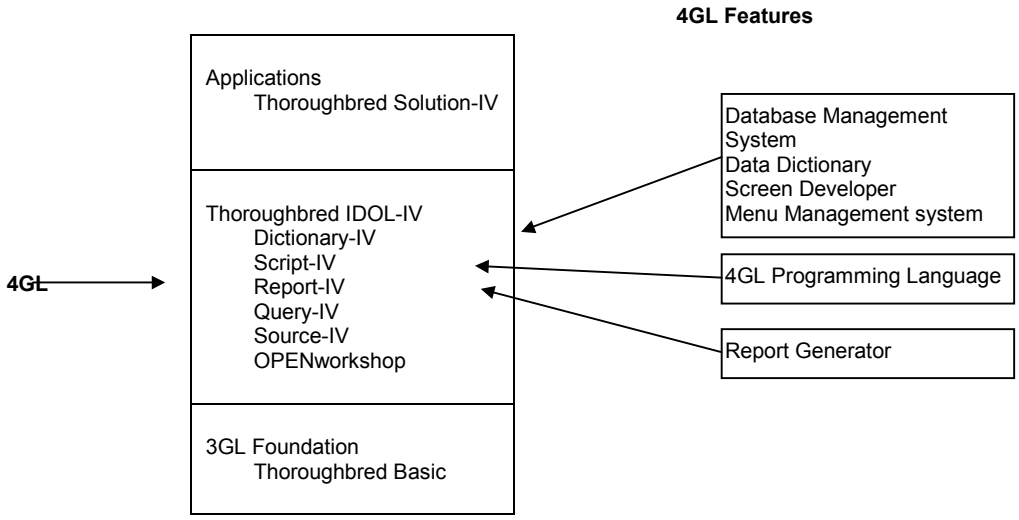
What is a Fourth Generation Language?

Fourth Generation Languages (4GLs) are often described as high-productivity languages, since this best describes their overall goal. These languages deliver this degree of improvement by providing a powerful set of development tools of which the programming language is an integral part.

A 4GL is not a language in the classical sense of programming languages. Instead, it is a system of integrated tools designed to be used for developing interactive, on-line, multi-user business applications. These development tools include:

- Database Management System
- Data Dictionary
- Screen Developer
- Menu Management System
- Report Generator
- 4GL Programming Language

4GL Development Tools



A 4GL enables rapid development of code and performs much of the general housekeeping associated with programming in a lower generation language. Thoroughbred IDOL-IV meets the definition of a Fourth Generation Language.

IDOL-IV

Thoroughbred IDOL-IV is a family of products that together provide a complete 4GL (fourth generation language) environment for the development of business application software.

Overview

IDOL-IV separates application logic from application data. The process executes 4GL Script-IV code and 3GL code (using Dictionary-IV API's) and makes reference to Dictionary-IV for all data items, data file definitions, screen designs, and user messages. These definitions are external to the application logic.

Components

Dictionary-IV
Is the system dictionary. It is a DBMS (Database Management System), which provides the foundation for the Thoroughbred IDOL-IV family of products. It also contains the public programs known as Dictionary-IV APIs.

Script-IV
Provides the 4GL procedural language constructs required for developers to build applications quickly. It uses structured, concise English-like commands that allow you to create applications using less code.

Report-IV
Is a comprehensive report generator. It runs with Dictionary-IV, which defines the type and location of data items. Report-IV combines data from multiple files, file structures, file systems, and operating systems. It can access applications across multiple machines.

Query-IV
Allows the software developer to place a query management system in the hands of the end-user. It is based on the industry standard SQL (Structured Query Language). Queries can be stored and retrieved later and can be transferred to a Report-IV format. Query-IV runs with Dictionary-IV, which defines the type and location of data items. The output may also be directed to Thoroughbred Gateway for Windows for further processing by any workstation-based application under Microsoft Windows.

Source-IV
Is a source code management system. Its editor has many advantages over the native Script-IV editor, including:

- An elegant interface with advanced editing functions.
- Source-code management capabilities.
- Edit histories that enable you to track and undo changes.

OPENworkshop
OPENworkshop is an object-oriented development environment. Object-oriented programming focuses on data objects and the operations that you perform on them. There is no longer a need to copy existing processes.

With OPENworkshop, Dictionary-IV applications are immediately applicable to object-oriented technology. The benefits of this technology are:

- Increased productivity in development.
- Increased productivity in maintenance.

There is not a main program. The information content and structure is implemented as Dictionary-IV Formats. The operations are implemented as Methods, which are procedural code written in Script-IV or Thoroughbred Basic and associated with the data object.

The CONNECT functions provide the interconnections between the data and the functional options available to the user.

- CONNECT Directive that allows for "connections" to be made from one object to another. A CONNECT may be made directly. For example: a Menu can CONNECT VIEW to display a defined View. It can also be made through a METHOD, which contains procedural code that modifies the behavior of the CONNECT. A CONNECT can be invoked at pre- or post- processing stages of data entry, or when defined function keys are used.
- METHOD Thoroughbred Basic or Script-IV code that can be associated with OPENworkshop objects. Typically, Methods will be associated with pre- or post-processing of data in Screens or Views, or may be invoked as I/O Triggers associated with a Link.

Overview of Thoroughbred Query-IV

Thoroughbred Query-IV makes use of a query language to help you get the information you need quickly. A query language contains words and symbols that you can use to request information from the database.

With Query-IV you do not have to use a query language, but instead let Query-IV create your query quickly and easily through selection windows.

If you prefer to use a query language, Query-IV uses a query language similar to SQL (Structured Query Language).

You can easily query a database and save and recall the query. This is a capability normally provided with embedded query languages, which allow you to embed queries in programs. Once you save a Thoroughbred Query, you can execute it from a Menu, Script (4GL program), or from Query-IV.

Database
A database is an organized collection of information or data. For example, a telephone book database would include Data Names like name, address, and telephone number.

SQL
The American National Standards Institute (ANSI) has defined a standard version of SQL based on the IBM version, and the International Standards Organization (ISO) has in turn defined an international standard based on the ANSI definition.

If you know SQL, Thoroughbred Query-IV will be familiar to you. If not you will find Thoroughbred Query-IV is easy to learn.

Tables and the Relational Database

A table is the result of a query. It contains information in vertical columns and horizontal rows, displayed in a two-dimensional, rectangular shape. A table can contain any number of columns and rows.

Each item of information in a column or row is called a field. A field is the intersection of a row and column.

The database that handles all information in terms of tables is known as a relational database. A relational database allows you to use tables without regard to how the information is physically stored in the computer. This makes it easier to get the information you need.

Query languages use a relational database. Tables, columns, and rows are relational database terms. If you are familiar with the terms describing a physical database, such as files, records, and Data Names, it may help to associate these with the terms describing a relational database.

Tables are files, rows are records, and columns are Data Names.

Although these terms are similar, they are not synonymous. One reason is that a relational database is a logical or conceptual structure that can be independent of the underlying physical database.

A table can represent information from part of a file or from more than one file.

A row can represent information from part of a record or from more than one record.

A column can represent information from part of a Data Name or from more than one Data Name.

Database

A database is an organized collection of information.

The telephone book is a database: the names, addresses, and telephone numbers are the data, and the names are organized alphabetically.

LAST-NAME	FIRST-NAME	STREET	CITY	TELEPHONE
Agassiz	Rick	21 Hill St.	Hot Springs	555-1889
Donato	Elyse	94 Red Rd.	Brookville	555-6391
Franklin	John	102 Main St.	Stover	555-3077
Sandlar	Penny	22 W. Elm St.	Tulsa	555-0425

An ordinary dictionary is also a database. The words and definitions are the data, and the words are organized alphabetically.

A checkbook register, in which you record check payments, is another database: the date, check number, payee, payment amount, and account balance are the data, and the payments are organized by date and check number.

The database does not have to be stored in a computer; it can be stored in an ordinary filing cabinet filled with file folders.

Thoroughbred IDOL-IV stores the information in computer data files. A database consists of one or more related files.

In the preceding examples, each database contained a single file. But a database can be as small or as large as you need it. The telephone database could contain additional files, depending on how much information you needed to control. This could be directories for all the areas in the state, all the states in the country, or all the countries in the world.

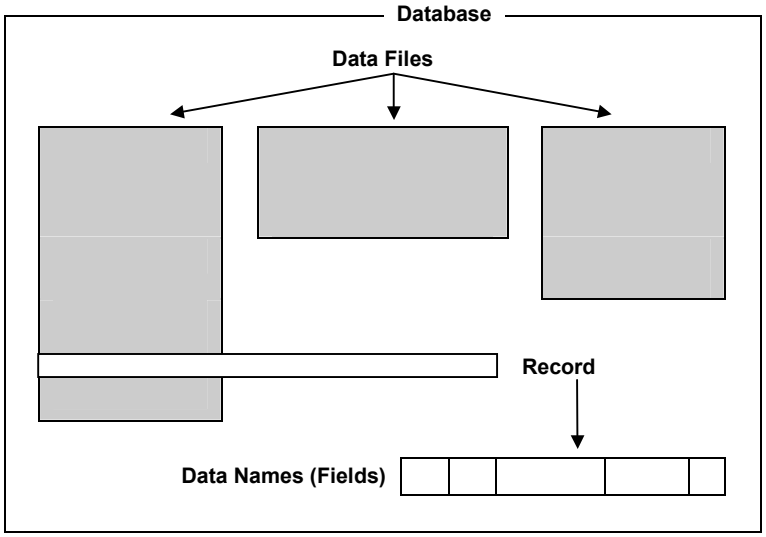
Files, Records, Fields
IDOL-IV references files, records, and fields. In SQL terms they are equal to the following:

IDOL-IV	SQL
Files	Tables
Records	Rows
Fields	Columns

One thing that makes a database so useful is that the information is organized. If the telephone book were not organized alphabetically, it would be a useless collection of information.

Organizing information requires that you use a structure, so the information can be stored away and located later as is needed.

The Database - Files, Records, and Data Names (Fields)



The database is the largest structure and can contain a number of related data files. We will discuss a small database with only one file.

The information in a database is organized into data files. A data file in the database would be the telephone file.

The information in data files is organized into records. A record in the telephone file example would be the information for one person: last name, first name, street, city, and telephone number. Records are shown as horizontal rows across the page.

The information in records is organized into Data Names (fields). The Data Names in the telephone file are the last name, first name, street, city, and telephone number. The Data Names are shown as columns.

Data Names are usually, but not always, the smallest pieces of data in the organization.

LAST-NAME	FIRST-NAME	STREET	CITY	TELEPHONE
Agassiz	Rick	21 Hill St.	Hot Springs	555-1889
Donato	Elyse	94 Red Rd.	Brookville	555-6391
Franklin	John	102 Main St.	Stover	555-3077
Sandlar	Penny	22 W. Elm St.	Tulsa	555-0425

Below is a listing of some common synonyms.

Data File	Table, Relation
Record	Row, Tuple
Data Name	Field, Column, Attribute

The information contained in each Data Name is referred to as a value. In the above example under the Data Name LAST-NAME, we have the following values: Agassiz, Donato, Franklin, and Sandlar.

A record has a standard format so every record in the file has the same Data Names. When you define a record, you define a Format, or the sequence of Data Names in the record.

Data Names have certain characteristics, including a name and attributes such as size, the type of data it can hold, and whether it is used as a key to organize the file. The following topics provide a basic understanding of Data Names.

Data Name and Size
When you create a Data Name, you give it a name and define the attributes. In the telephone file example, you might name one Data Name LAST-NAME and give it a size of 20. Values entered in this field cannot exceed 20 characters.

The following example is a 20-character field.

Robinson

R	o	b	i	n	s	o	n												

Rosencrantz-Guildenstern

R	o	s	e	n	c	r	a	n	t	z	-	G	u	i	l	d	e	n	s

One reason for having a fixed size for Data Names is so your system can find information very quickly. Since Data Names have a fixed length, every record has a fixed length. No matter which record you access, the Data Names can be found in the same position.

Types of Data
Information comes in many forms. A name is information and so is a word, a date, a number, or an amount of money. Each type of information is handled differently. A number can be added, subtracted, and multiplied with another number, but not so with a person's name.

DATA-NAME	Type	Data
LAST-NAME	Text	Krupchezki
QTY-ON-HAND	Numbers	9301
ORDER-DATE	Dates	3/31/97
ORDER-TOTAL	Money	$1,532.09
CUSTOMER-PHONE	Telephone	732-555-0302
EMPLOYEE-SS	Social Security	911-00-9019

Key Fields
Keys are Data Names that help you to organize and locate information in a file. In the telephone file, the LAST-NAME is a key field, because the file is in order by last name. If you want to locate Sam Jones in the telephone file, you look up Jones.

Keys help to identify the information for you. But, for most files, the system needs a special key so that it can identify information: the primary key.

Primary Keys
The primary key is a unique identifier for the record. If everyone in the telephone book had a different last name, then you could use LAST-NAME as the primary key. But since this is not likely to be the case we must use some other method to uniquely identify each person.

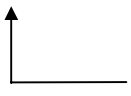
If the people with the same last name all had different first names, then you could use the LAST-NAME plus the FIRST-NAME. In this case, the primary key would consist of two key fields: LAST-NAME and FIRST-NAME.

The primary key must contain a unique value that is the name of one and only one record. A social security number is a good example of a unique identifier.

Sometimes it is easier to invent a primary key. For example, you might create a Data Named code that contains values from 0001 to 9999.

The primary key is often informally called the key of the file; however, the key may contain one or more primary key fields. In this example the company name is the primary key.

Primary Key Order: COMPANY Name				
COMPANY	**STREET**	**CITY**	**PHONE**	**BALANCE**
Boots & Saddles	21 Hill St.	Hot Springs	555-9561	$203.00
Chamois by Kate	94 Red Rd.	Brookville	555-1352	$174.59
Great Rider	102 Main St.	Stover	555-0880	$962.05
Patently Leather	22 W. Elm St.	Tulsa	555-0465	$564.23



In order by COMPANY name (ascending alphabetically).

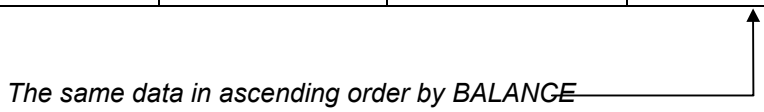
The primary key is used to organize the file and to locate information. You may also use secondary keys to help you locate information that is not in the primary key.

Secondary Keys
Secondary keys allow you to access a file in an order other than primary key order. Unlike primary keys, secondary keys in IDOL-IV do not have to be a unique identifier.

Secondary keys are also known as cross-indexes, or sorts.

In the following example the BALANCE is used as the secondary key field. The same data is listed in ascending order by balance:

Secondary Key Order: BALANCE				
COMPANY	**STREET**	**CITY**	**PHONE**	**BALANCE**
Chamois by Kate	94 Red Rd.	Brookville	555-1352	$174.59
Boots & Saddles	21 Hill St.	Hot Springs	555-9561	$203.00
Patently Leather	22 W. Elm St.	Tulsa	555-0465	$564.23
Great Rider	102 Main St.	Stover	555-0880	$962.05



Software Conventions

The following sections describe IDOL-IV software conventions.

Screen Navigation

If you are running VIP (Visual Image Presentation) Gateway for Windows, you may use your mouse to click and double-click on your selections as you normally would in the Windows environment.

If you are not running VIP Gateway for Windows, you must move through your character-based application by using the **Arrow Keys**, **Tab Key**, **Function Keys**, and **Character Keys** (**0-9**, **A-Z**, etc.).

Function Keys

Function keys available in all Thoroughbred products:

F4 Ends an operation or exits a function. The system allows you to save your addition or change or exit without saving.

F6 Displays on-line help in a window.

Selection windows

Thoroughbred Dictionary-IV is controlled through a series of windows. These windows employ a graphical interface, providing an easy method for using Thoroughbred Dictionary-IV.

To choose an option in a selection window, highlight the item and press **Enter**. If you select an item by mistake, press **F4** to cancel the selection.

Menu Types

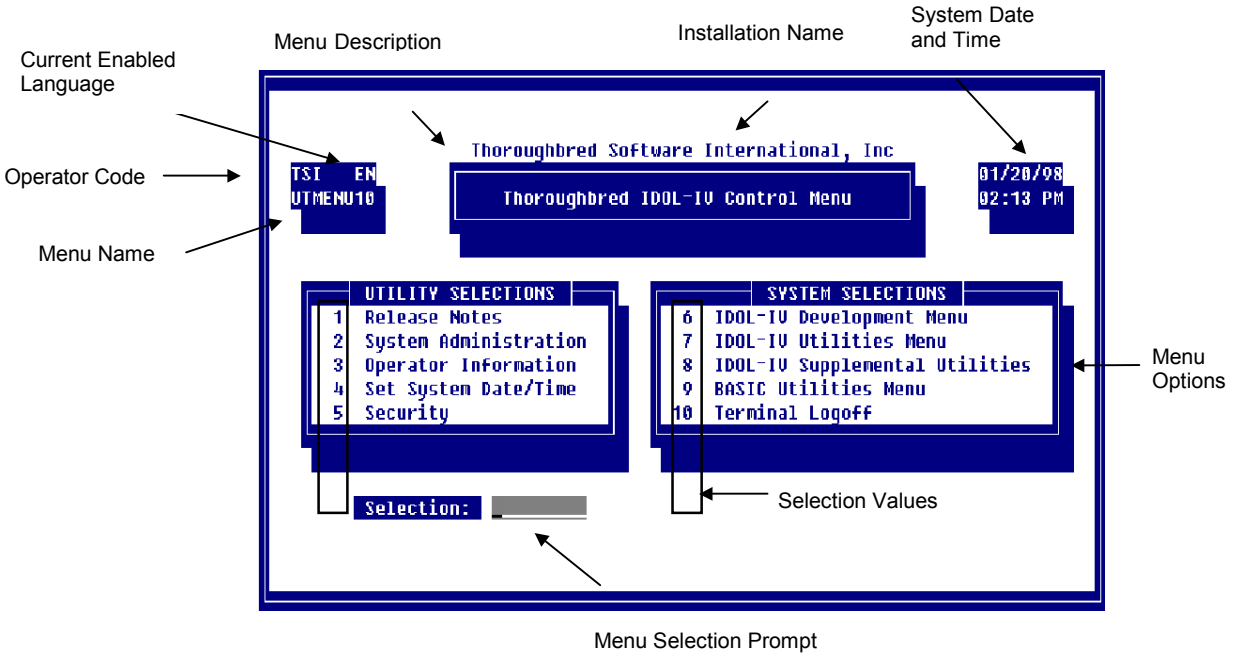
In IDOL-IV there are three different types of menus:

- IDOL-IV
- Pop-up
- Toggle (character-based applications)

Each is described below:

IDOL-IV
A menu screen is used to display options. You must type a selection and press **Enter**. For example type **UTMENU10** and press **Enter** to see the following:

IDOL-IV Control Menu

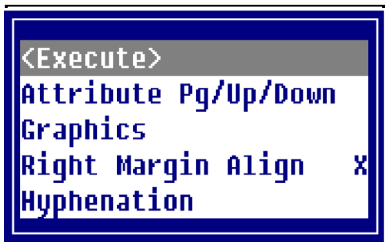


Pop-up
Pop-up menus allow you to scroll with a lightbar to highlight an item and press **Enter** to select it. For example press **F1** from any IDOL-IV menu to see the following pop-up menu:

You may select options by:

- Typing the first character of the line and pressing **Enter**. Type **4** and press **Enter** to select Link definition.
- Using the **Down Arrow** or **Up Arrow** keys to highlight the option and pressing **Enter**.
- Using the **Home** key to toggle between the first and last option in the list.
- Using the **Page Up** key to move to the first option in the list, or **Page Down** key to move to the last option in the list.

Toggle
Toggle menus are used in character-based applications and allow you to select one or more items on the menu. Press the **Space bar** or **Enter** key to toggle.



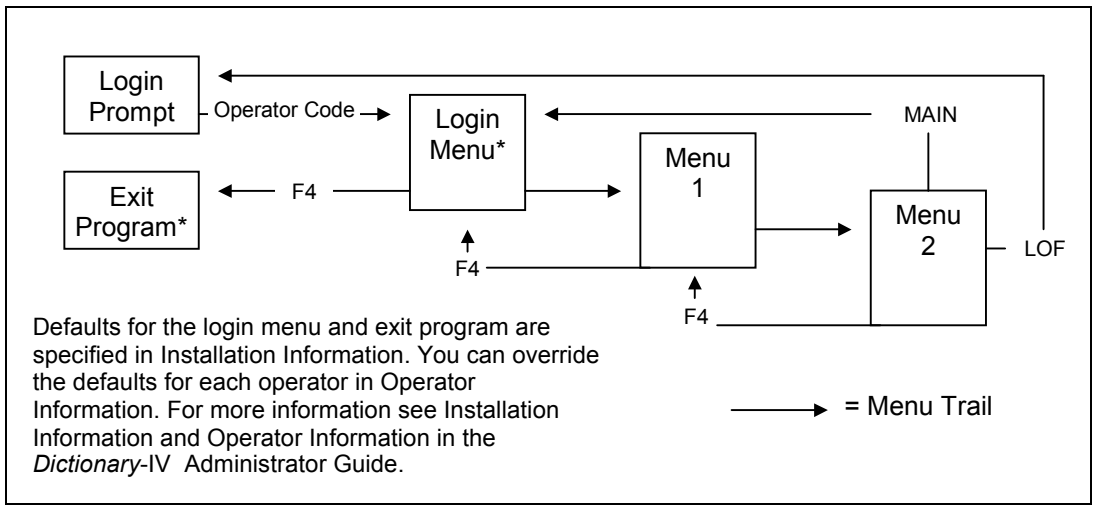
Exit and apply changes.
Page Up/Down disabled.
Graphics mode disabled.
Right margin text alignment enabled.
Hyphenation

You must select **<Execute>** to exit and save your changes. Press **F4** to exit without saving and enabling your selections.

IDOL-IV Menu Trails

A menu trail is created as you move from menu to menu in an application. For example, selecting the IDOL-IV Development Menu from the IDOL-IV Control Menu and the IDOL-IV Utilities Menu from here, sets up a menu trail. Pressing the **F4** key always takes you to the previous menu in the menu trail.

Menu Processing



Printing

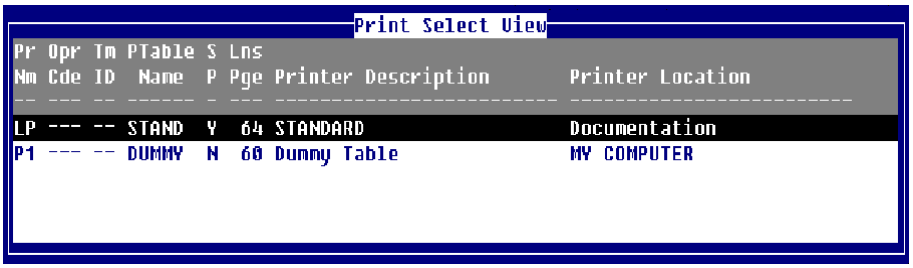
If you select a print option or press **F16**, IDOL-IV prompts:

`Select Printer (xx, x1x1, x2x2):`

xx, x_1x_1, x_2x_2 refers to the names of the printers you added during Printer Assignment. For more information, see Printer Table Maintenance and Printer Assignment in the Dictionary-IV Administrator Guide.

Type the printer name and press **Enter**. Press **F1** for a view of printers:

F1 Print Select



The following values are defined in **Printer Assignment**. For more information, see Printer Assignment in the Dictionary-IV Administrator Guide.

Pr Nm
Printer Name (LP, P1, P2, etc.).

Opr Cde
Operator Code that has this printer open.

Tm ID
Terminal ID that has this printer open.

P Table Nam
Printer Table Name.

SP
Spooled (yes/no). Select **Yes** to send the output to a print spooler.

Lns Pge
Number of lines per page.

Common Syntax Elements

Some commands in this manual use common syntax elements, which are explained in this section.

Expressions
An expression is two or more elements, consisting of constants, Data Names, variables, or functions that interact with operators to form a new value.

An expression can be used in a screen data field (formula), and conditions.

Operators in Expressions

+	addition or string concatenation
-	subtraction
*	multiplication
/	division
^ or **	exponentiation
( )	grouping

Numeric expressions are evaluated according to the normal priority convention (grouping, exponentiation, multiplication/division, addition/subtraction).

Logical Conditions
A logical condition is two or more values, consisting of constants, Data Names, variables, functions, or expressions that interact with relational or logical operators to form either a true or false result.

Operators in Conditions

=	equal to
>	greater than
<	less than
>= or =>	greater than or equal to
<= or =<	less than or equal to
< > or > <	not equal to
( )	grouping
AND	logical AND (both true)
LIKE	search for a match using partial values
OR	logical OR (either true)

String
A string is a connected sequence of characters treated as one piece of data.

Substring
Substrings can be used in screen data fields, Sort definitions, and view commands. They allow you to specify a portion of the original value in a Data Name.

- Substrings can only be used with alphanumeric Data Names.
- Substrings cannot be used with a multiple occurrence Data Name.

The syntax is as follows:

```
string-elmt[(start,length)]
```

`string-elmt` is an alphanumeric Data Name.

`start` is the starting position for the substring.

`length` is the number of characters of the substring.

Examples:

CUST-NAME(1,5)
This specifies the first 5 characters of the CUST-NAME Data Name.

ITEM-CODE(1)
This syntax is valid only when ITEM-CODE is a multiple occurrence, and it does not specify a substring. This specifies the entire value of the first occurrence in a multiple occurrence Data Name. For more information, see Data Name and Multiple Occurrence in the Dictionary-IV Developer Guide.

Masks (Output Formats)

A mask specifies the format in which numeric data is displayed or printed. A mask can be used in screen data fields, reports, queries, and scripts.

Masks allow you to specify the same output format regardless of the size of the number. They also allow you to output financial symbols and characters (for example to format 563204 as $5,632.04-), and to retain insignificant zeros (for example to output 1 as 1.00).

Syntax
The syntax is as follows:

num-elmt:"mask"

num-elmt is a numeric Data Name, formula, or multiple occurrences.

mask is one or more mask characters.

Mask Characters
Each digit in the numeric value that is output must match up with a mask character. Mask characters determine the output. You can specify more mask characters than there are digits in the numeric value.

An insignificant zero is a leading or trailing zero. A floating character is output at the right-most leading zero.

0 Outputs a digit from 0 to 9. When the digit is an insignificant zero, it outputs a 0.

Outputs a digit from 0 to 9. When the digit is an insignificant zero, it outputs a space.

. Outputs a decimal point between digits. If the number to the right is 0 and 0 is not used in the mask, it outputs a space.

, Outputs a comma between digits if the digit to the left is a significant digit; it can only be used to the left of the decimal point.

$ Outputs a dollar sign.

***** Outputs an asterisk when the digit is a leading zero.

- Outputs a minus sign if the number is negative, or a space if the number is positive. This mask character can be placed at the left or right of the mask. If placed to the left, it is a floating character.

+ Outputs a plus sign if the number is positive, a minus sign if the number is negative, or a space if the number is zero. This mask character can be placed at the left or right of the mask. If placed to the left, it is a floating character.

CR Outputs a CR (to indicate a credit) if the number is negative, or 2 spaces if the number is positive. This mask character must be placed at the right of the mask.

DR Outputs DR (to indicate a debit) if the number is positive, a CR if the number is negative, or a space if the number is 0. This mask character must be placed at the right of the mask.

(Outputs a left parenthesis if the number is negative, or a space if the number is 0 or positive. This mask character must be placed at the left of the mask.

) Outputs a right parenthesis if the number is negative, or a space if the number is 0 or positive. This mask character must be placed at the right of the mask.

B Outputs a space.

???? Outputs any characters other than the mask characters and places them in the specified position.

How This Manual is Organized

The Thoroughbred Query-IV Reference Manual is organized as follows:

Introduction
Provides an overview of IDOL-IV and Query-IV. Describes software and documentation conventions.

Using Query-IV
Describes how to set up and use Query-IV. Shows how to login and log off and provides a summary of the Query-IV menus and options.

Operator Preferences
Describes how to set Query-IV Operator Preferences.

Query Framer
Describes how to create and execute the queries.

EQL Command Language
Defines the syntax for EQL, as an SQL compatible query language.

Query Maintenance
Describes the maintenance features for Query-IV. Describes the use of Schemas that use multiple databases to create a query.

Query Execution
Describes how to execute a query and display a table of information requested by the query.

Messages
Lists and defines Query-IV/IDOL-IV error messages.

Sample Queries and Databases
Provides sample queries and databases as examples for using Query-IV.

Glossary
Provides a glossary of terms used in the IDOL-IV environment.

Documentation Conventions

The following sections describe IDOL-IV documentation conventions.

Keyboard Designations

Keys on the keyboard are identified in bold type. For example:

Press **Enter** to make your selection or press **F4** to exit.

The documentation refers to **Line-Insert Key**, **Line-Delete Key**, **End-of-Line Key**, etc. These are the functions of particular keys. Please refer to your keyboard documentation to determine the specific key that performs this function for you.

Prompts and Messages

Software prompts and messages are indented and shown in bold type. For example:

```
RETURN-Select, F4-END
Save Changes (Y/N)
```

Prompts and messages may refer to Return,<CR>, or Carriage Return. This manual refers to these as the **Enter** key.

Responses

Options you may select or information you must type appear in bold type. It must be entered exactly as shown. For example:

Type **John Smith** and press **Enter**.

Notational Symbols

Syntax appears in a bold non-proportional font to better display spaces and punctuation. Elements of the syntax are always explained directly below. The following conventions are used in command formats throughout the Thoroughbred product line.

Command Syntax

CONNECT VIEW {***view-name-1***}[[***view-name-2***]...]

Notational Symbol	Description
	CONNECT VIEW
(BOLD FACE/UPPERCASE)	Commands or keywords you must type exactly as shown. Example: CONNECT VIEW UTVCUST You must type CONNECT VIEW including the space.
	view-name-1
Italic Face	Information you must supply. Example: CONNECT VIEW UTVCUST You supply the view name UTVCUST. In most cases, lowercase italics denotes values that accept uppercase or lowercase characters.
	{*view-name-1*} [[*view-name-2*]...]
Braces { }	You must select one of the values enclosed by the braces. Example: {UTVCUST1} [[UTVCUST2]... You must supply the view name UTVCUST1.
Brackets [ ]	You can select one of the options enclosed by the brackets; none of the enclosed values is required. Example: {UTVCUST1} [[UTVCUST2]... You must supply the view name UTVCUST1. You may also supply the optional view name UTVCUST2.
Ellipsis …	You can repeat the word or clause that immediately precedes the ellipsis. Example: {UTVCUST1} [[UTVCUST2]... You must supply the view name UTVCUST1. You may also supply the optional view names UTVCUST2, UTVCUST3, UTVCUSTX. X represents any existing view name in the library.

Other Notational Symbols

Notational Symbol	Description
	Option-1 | Option-2 | Option-3
Vertical Bar |	Piping separates options. One vertical bar separates options, two vertical bars separate three options. You select only one of the options. Example: Add | Change | Delete You may select the Add, Change, or Delete option.
Underscores *or UPPERCASE ITALICS*	Either a default in a command description or display default in a screen image. Example: Add | Change | Delete You may select the Add, Change, or Delete option. The Change option is the default value.

Thoroughbred Product Line

The following briefly describes other Thoroughbred products.

Basic
Basic is the Thoroughbred version of Business BASIC. It includes a library of functions that provide access to the facilities of Dictionary-IV from Thoroughbred Basic programs called Dictionary-IV APIs.

VIP/Gateway for Windows
This is a family of software products that enables Microsoft Windows workstations to access host-based applications. It also allows host applications to control a Windows workstation application.

VIP (Visual Image Presentation) for Dictionary-IV allows any application that has been developed under Dictionary-IV to provide a GUI (Graphical User Interface) Microsoft Windows workstation. Not one line of the application code needs to be changed to add a GUI to a character-based application.

Gateway for Windows allows any supported 3GL or 4GL host application to communicate with any application running on a PC under Microsoft Windows. You can build host-based applications that communicate with Windows-based application software using DDE (Dynamic Data Exchange). And you can exchange data between those Host and Windows applications in either direction.

You may use products such as Microsoft Excel to display charts and graphs that update dynamically. Graphics are redrawn in real time.

Solution-IV Accounting
Solution-IV Accounting is a completely integrated multi-user, multi-company accounting solution. It consists of: General Ledger, Accounts Payable, Accounts Receivable, Bank Reconciliation, Fixed Assets, Inventory Control, Order Processing, Purchase Order, and Payroll.

Using *Gateway* for Windows you can create and display Microsoft Excel graphs and charts and insert data into Microsoft Word documents. Using an image management package, both color and scanned document images can be attached to *Solution*-IV data.

Solution-IV complies with GAAP (Generally Accepted Accounting Principles).

TS Data Server for Networks, ORACLE, and ODBC
Thoroughbred environments support most of the leading database systems. This means you can design an application once and implement it on a wide variety of databases without change.

In particular, Thoroughbred provides different drivers for each supported Server environment. These drivers handle network protocols and translate data access requests into API calls or SQL statements, as appropriate. The driver manages the execution of these calls and returns the result to the Client application.

All of this occurs completely transparent to the Client application, and no code changes are required to change the application from one Server environment to another.

- Export to Microsoft Windows. When combined with the capabilities provided by Gateway for Windows, you can develop Server applications that are completely open to Microsoft Windows, and Microsoft NT applications.

- Import from Microsoft Windows. Thoroughbred IDOL-IV now provides ODBC (Open Database Connectivity). With Dictionary-IV installed on the Host, you can access IDOL-IV databases from any workstation across your networks, and import them into your application. You can, for example, import IDOL-IV data into your Microsoft Word application as a table. You can then display the data in any way supported by Microsoft Word and save it as a .doc file.

Services Provided by TSI

The following sections outline the services provided by Thoroughbred Software International, Inc.

On-line Documentation

Thoroughbred provides on-line documentation for IDOL-IV products. You must have installed the Developer Reference module during IDOL-IV installation. For more information, see the IDOL-IV Installation Guide.

You may access Help by pressing **Ctrl-P** and selecting **Help Topics** or by pressing **F6** at any menu option or any field.

The **Help Topics** option allows you to search the Help by a keyword or phrase. For example, if you need help for the Sort File field in Link Definition type **Sort File** and press **Enter**. The system displays all matching Sort file topics.

Support

Thoroughbred offers customer support by telephone, fax, and e-mail. Our e-mail address for customer support is **support@tbred.com**. Contact your sales representative for more information.

Thoroughbred On-line

Thoroughbred provides information and services on the Internet. Please visit Thoroughbred on the World Wide Web at **http://www.tbred.com**.

Thoroughbred publishes information and responds to comments and questions through this service. It provides a 24-hour reference and communications vehicle for Thoroughbred developers and end-users worldwide.

Training Classes

Thoroughbred offers training classes on IDOL-IV and other products. These classes are held at the corporate headquarters as well as regional locations. For more information, and a schedule of upcoming classes, please contact your sales representative at 800-524-0430 or 732-560-1377.

For More Information

... about creating applications, refer to the Thoroughbred Dictionary-IV Getting Started Guide.

... about using each IDOL-IV product, refer to the individual reference manual (Query-IV, Report-IV, Script-IV, and Source-IV Reference manuals).

... about Thoroughbred Basic Utilities refer to the Thoroughbred Basic Utilities Reference Manual.

... about Thoroughbred Basic, refer to the Thoroughbred Basic Reference Manual.

USING QUERY-IV

Before you can begin to create queries using IDOL-IV, your system must be properly configured. Check with your System Administrator before you begin.

Procedure for set up

This section will take you through the steps you should follow to create queries.

Login

IDOL-IV must be running. You can start Thoroughbred IDOL-IV using one of the following methods:

- From the Thoroughbred Basic Utilities Menu, type **ID** and press **Enter**.
- From Thoroughbred Basic Console Mode, type **RUN"ID"** and press **Enter**.

As a default the system displays the Thoroughbred IDOL-IV copyright screen.

> The login menu is defined in Operator Information and determined by the System Administrator.

Press **Enter** to continue to the Operator Login Screen. The system prompts:

```
Thoroughbred IDOL-IV - Enter Operator Code:_____
```

Type your operator code and press **Enter**. See your System Administrator for your Operator Code.

Once IDOL-IV is running you may execute Query-IV as follows:

- Type **TQ** at any IDOL-IV menu and press **Enter**. The system displays the Query Control Window.
- Type **RUN"Q4GO"** from Thoroughbred Basic Console Mode and press **Enter**. The system displays the Query Control Window.
- Type **Q4GO** from the Thoroughbred Basic Utilities Menu and press **Enter**. The system displays the Query Control Window.
- Press **F1** from any IDOL-IV menu and select **Query**. The system displays the Query View. To display the Query Control Window from the view press **F1–Edit**. To perform maintenance within the view, refer to the Dictionary-IV User Guide.

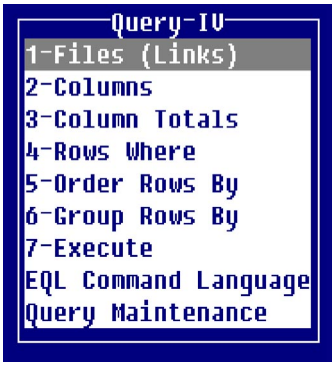
Set Operator Preferences

This function allows you to set such defaults as query description, output device (terminal/printer), and library name. It also allows you to set the exit program name. For more information, see the Operator Preferences chapter later in this document.

Create and Maintain the Queries

To create and maintain queries use the Query Control Window. To access the Query Control Window see the Login section above.

Query Control Window with Query Framer



The following function keys are available from the Query Control Window:

Enter To select an option.

F4 To exit.

F6 To display on-line help.

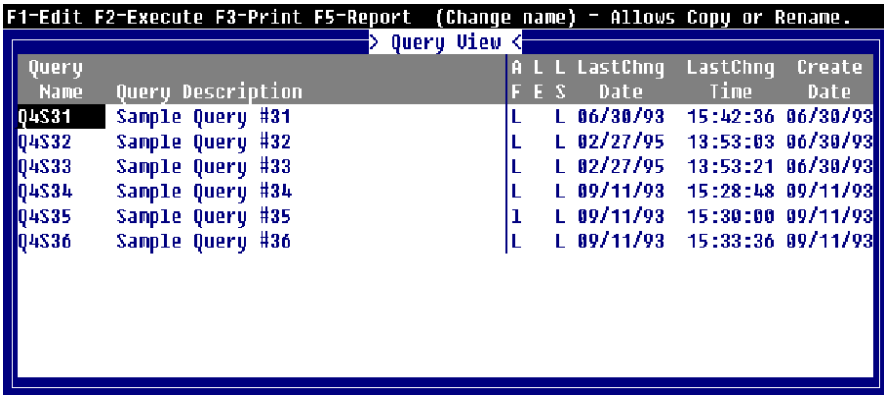
F7 To select Special Functions.

The Query Control Window allows you to create, maintain, and execute queries. It also allows you to access the EQL Command Language (SQL compatible language) for editing Queries. For more information, see the Query Framer and EQL Command Language sections later in this document.

The following description briefly shows how to frame a query, run it, save it, and exit Thoroughbred Query-IV. More examples can be found in the Sample Queries and Databases section of this document.

Every query must specify something from the first two selections in the Query Framer: Files (Links) and Columns. All other selections are optional.

If you accessed Query-IV by pressing **F1** and selecting **Query** the system will display the Query View:



To access the Query Control Window from the Query View press **F1-Edit**. For information on maintenance within the view, refer to the Dictionary-IV User Guide.

Select a Link
Select **Files (Links)**. The Link selection window displays. Press **F10** and the system prompts:

```
Enter key value:
```

Type **Q4EMPLOY** and press **Enter**. The system highlights the Link name that you typed. Press **Enter** to select the Link name. It displays in the Files (Links) window. Press **F4** to exit the Link selection window.

Select Columns
Select **Columns**. The column selection window displays. Select **Employee Name** and **Job Description**.

Press **F4** to exit the column selection window.

Execute the Query
Select **Execute**. The table that you requested in your query is displayed. Assuming that you selected the columns **EMPLOYEE-NAME** and **JOB-DESCRIPTION**, this is what the resulting query table looks like:

Employee Name	**Job Description**
Douglas	Bookkeeper
Roland	Programmer
Bell	Marketing Rep.
Parker	Programmer
Watson	Marketing Rep.
Alexander	Sales Rep. (Regional)
Drummond	Programmer
Johnson	Sales Rep.
Mason	Sales Rep. (Regional)
Brady	Sales Rep.
Perkins	Sales Rep.
Allen	Sale Rep. (Regional)
Davis	Bookkeeper

```
End of Report. <CR> to continue.
```

Press **Enter**, and the Query Control Window displays.

The EQL Command
When you create your query through the Query Framer Thoroughbred Query-IV creates an EQL command from your selections.

To see the EQL command select **EQL Command Language** from the Query Control Window. This displays the EQL window. Press **F9** to generate the EQL command from the Query Framer selections.

The syntax of the EQL command is:

```
SELECT columns FROM link
```

The EQL command in this example is:

```
SELECT EMPLOYEE-NAME, JOB-DESCRIPTION FROM Q4EMPLOY
```

Once you learn Thoroughbred Query-IV, you can directly enter, edit, and execute queries in this EQL Command Language window. Press **F4** to return to the Query Control Window.

Save the Query
You can use Thoroughbred Query-IV simply to examine the database without saving your queries. However, you can save the query either while you are in Thoroughbred Query-IV or when you leave.

- To save the query while you are in Thoroughbred Query-IV, select **Query Maintenance** from the Query Control Window. The Query Maintenance Window displays. Select **New Query (Clear)**. Whenever you select New Query (Clear) or Retrieve Query, you may save the current query.

- To save the query when you exit, press **F4** and you may save the current query.

Logoff

First press **F4** until you return to IDOL-IV. There are several ways to exit Thoroughbred IDOL-IV.

To return to the Operator Login Screen:
Type **LOF** at any IDOL-IV menu and press **Enter**.

To return to the operating system:
Select the **Terminal Logoff** option at the IDOL-IV Control Menu or the IDOL-IV Development Menu.

To return to the Operator Exit program:
Press **F4** at any IDOL-IV menu until you reach the initial menu. When you reach this menu press **F4** to run to the operator exit program. You may define this exit program using the Operator Information utility. For more information, see Operator Information in the Utilities section of this document.

To return to Thoroughbred Basic Utilities:
Select the **BASIC Utilities Menu** option from the IDOL-IV Control Menu.

To run another Thoroughbred Software product:
Press **F3** and make a selection. When you run one of these products from IDOL-IV, you are automatically logged in with the current operator code.

To run a program:
Type a slash (/) or a colon (**:**) followed by the program name and press **Enter**. Examples: **/DESKMGR** or **:DESKMGR**.

To run another menu:
Type **;menu-name** and press **Enter**. The menu-name is the 3 to 8 alphanumeric character name of the menu to display. Examples: **;UTMENU10** or **;UTMENU22**.

Timeout:
Exits to the Operator Login Screen. Timeouts occur when you leave a terminal unused at an IDOL-IV menu. The amount of time before logoff can be set in the System Administration options.

Thoroughbred Query-IV Menus

In this section the Query-IV menus are briefly described. For more information, see the appropriate section in this document.

The following menus are provided in the Query Control Window:

- Query Framer
- EQL Command Language
- Query Maintenance

Query Framer

This menu is used to create and execute queries.

Files (Links)
Selects the File or Link to use to retrieve data for this query. You select one of these before doing anything else.

Columns
Selects the Data Names from the File or Link to be used for this query. You must also select from this option before doing anything else.

Column Totals
Totals numeric columns.

Rows Where
Sets conditions for selecting data from the File or Link.

Order Rows By
Allows you to select a sort or create a temporary sort. You cannot Order Rows By and Group Rows By in the same query.

Group Rows By
Groups rows together by the specified Data Name and produces a summary row for each group. You cannot use Order Rows By and Group Rows By in the same query.

Execute
Executes the query.

EQL Command Language

The EQL Command Language window allows you to create and edit an SQL-compatible query language.

Query Maintenance

Allows you to edit queries and schemas.

Retrieve
Retrieves the selected query.

Delete
Deletes the selected query.

Rename
Renames the selected query.

Copy
Copies the selected query.

List
Lists a defined range of queries to the terminal or a printer.

Print Definitions
Prints copies of the actual query definition, containing all of the specifications, content, and Format information.

New Query (Clear)
Allows you to create a query from a clean slate not a copy of another query.

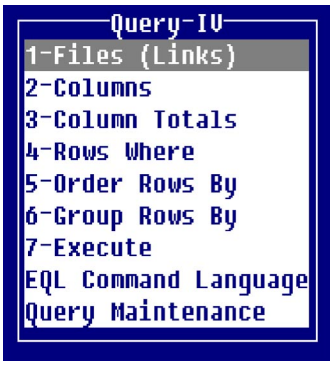
Query Parameters
Directs you to the method for viewing or editing operator parameters.

Schemas
Allows you to create and edit queries that use more than one database.

OPERATOR PREFERENCES

When you run Thoroughbred Query-IV the system displays the Query Control Window:

Query Control Window



If you access Thoroughbred Query-IV by pressing **F1** and selecting **Query** the system displays the Query View.

From the Query Framer the following function keys are available:

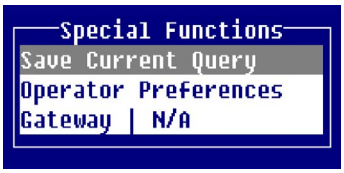
Enter To select an option.

F4 To exit the Query Framer.

F6 To display on-line help.

F7 To access the Special Functions.

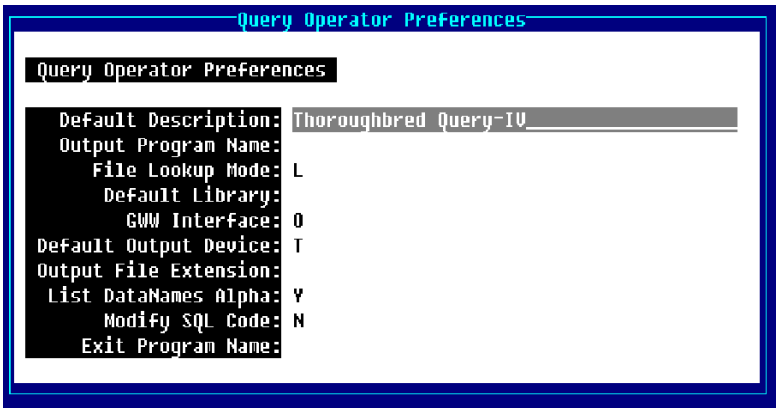
Press **F7** to access the Special Functions.



Select **Operator Preferences**.

The system displays a screen similar to the following:

Operator Preference



You may enter the following information:

Default Description
Type up to 30 alphanumeric characters for the description and press **Enter**. This heading displays when the output is to a printer or the EQL Heading is on. For more information, see the Output Options (Heading) section of the EQL Command Language section in this document.

Output Program Name
Type the 3 to 8 alphanumeric character program name. This is program to be executed when the query output is directed to a file that will be executed prior to printing any detail to the file.

You can use it to manipulate detail into a specific file format. The output program should be a public script that receives two variables. The first is a numeric containing the output channel. The second is a string containing the detail record.

File Lookup Mode
Allows access to Dictionary definitions through Links or Schemas. When in the Link mode you have access to all Links. When in the Schema mode you only have access to defined Schemas.

L Displays only Links (single files) when you select **1-Files** (Links). When you request Data Names, the system displays Data Names for Links.

S Displays only Schemas (predefined databases) when you select **1-Files** (Links). When you request Data Names, the system displays Data Names for Schemas.

For more information about Schemas, see the Query Maintenance section of this document.

Default Library
Type the 2 alphanumeric character library name to be used as a default for Link name selection and press **Enter**. Once Links are selected, the last Link name will be used as a pointer.

This option is only applicable for creating new queries. If no name is specified, Query-IV will start with the first non-numeric library.

GWW Interface
Allows you to select the GWW start flag as on or off. Press **Enter** select one of the following:

N To turn the GWW interface on.

O To turn the GWW interface off.

If GWW has not been started the flag will be set to N/A for the session.

This interface displays when you press **F7** Special Functions at the Query Control Window.

Default Output Device
Allows you to select File, Printer, or Terminal as the default output device for your queries.

Select one of the following:

F Query-IV will output the results to a file. The name of the file will be the same as your query with an optional extension. This extension comes from the Output File Extensions field (see below). You may select an Output Program Name.

P Query-IV will prompt you for a printer each time you execute. The Default Description displays on these queries.

T Query-IV will output the results to your terminal.

Output File Extension
Type the characters for the default extension to be added and output to a file and press **Enter**. The file name for output will be the name of the query plus an extension.

If no extension exists, your operator code will be used.

Example:

Q4SAMPLE.TXT
Q4SAMPLE.WKS
Q$SAMPLE.DIF

Where Q4.SAMPLE is the query name and TXT, WKS, and DIF is the output extension.

List DataNames Alpha
Allows you to select the order in which data names are displayed. Press **Enter** to select alphabetical or physical.

Select one of the following:

Y Whenever the system displays Data Names in Query-IV, they will list in alphabetical order.

N Whenever the system displays Data Names in Query-IV, they will list in physical order. This is the order of entry in Format definition.

Modify SQL Code

Allows developers the option to modify the generated SQL code. Prior to syntax checking and execution, if this flag is set to **Y**, Query-IV passes the current SQL request to a public script to make any modifications that are necessary. Modifications at this level are often related to the WHERE condition.

When this option is used, Query-IV executes:

```
RUN PUBLIC "Q4WHERE",Q0$
```

Q0$ contains the current SQL request.

Select one of the following:

Y Allows you to modify SQL code.

N Does not allow you to modify SQL code.

If you select **Yes**, Query-IV will pass the current SQL request to a public script. Here you can make any modifications that are necessary prior to syntax checking and execution.

When this option is used, Query-IV will execute:

```
RUN PUBLIC "Q4WHERE",Q0$
```

Q0$ contains the current SQL request.

> The developer is responsible for creating and maintaining the 4GL public script Q4WHERE.

Exit Program Name

Type up to 8 alphanumeric characters for the name of the program to execute when exiting Query -IV. Press **Enter**. To return to an IDOL-IV menu, use the default program name ID. To return to the operating system type the value [SYSTEM].

The default is **ID**.

QUERY FRAMER

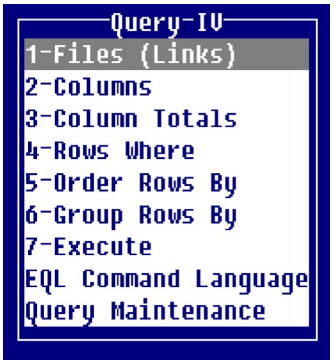
The Query Framer allows you to define a query, or frame a request, without having to use EQL Command Language. To create a query select from the options presented in the Query Framer windows. The Framer automatically generates the query command for you.

You can then execute the command generated by the Query Framer or bring the command into the EQL window for further editing and changes. For more information, see the EQL section in this document.

Executing the Query Framer

When you run Thoroughbred Query -IV the system displays the Query Control Window:

Query Control Window - Query Framer



Selections 1 through 7 of the Query Control Window are known as the Query Framer. Selections 1 through 6 display selection windows that you may use to create queries. Selection 7 executes the query in the Query Framer.

If you want to change or execute an existing query, select the Query Maintenance option in the Query Control Window. After you retrieve an existing query through Query Maintenance, you can use the Query Framer to change or execute it.

You are required to select something from the first 2 selections in the Query Framer: Files (Links) and Columns. All the other selections are not required.

The following keys are active in the Query Framer:

Enter Executes the highlighted selection.

F4 Exits the Query Control Window.

F6 Displays Help for the highlighted selection.

F7 Special Functions.

Up Arrow Moves up one selection.

Down Arrow Moves down one selection.

Home Moves to the top of the window.

[X] Moves to the selection beginning with the character you type (1, 2, 3, 4, ..., a, b, c, d, ...).

F7 Special Functions

The following options are available in Special Functions:

Save Current Query
Select this option to save the current query and then continue working on this query.

Operator Preferences
Select this option to set Query-IV Operator Preferences. For more information, see the Operator Preferences section of this document.

Gateway
If the Gateway interface is active this allows you to turn Gateway access on or off. It displays N/A if Gateway is not installed.

Files (Links)

This option selects the files to be used in creating this query.

The system displays a listing of links or schemas. You may define this mode in Query-IV Parameters (Operator Preferences).

> You must select at least one Link or Schema before selecting any other options in the Query Framer.

The selection window displays a list of Links or Schemas. More than one may be selected, but they must be related. When using Schemas, you may only select one.

Your selection is placed in a window and you may continue to make selections or press **F4** to continue creating the query.

> In EQL, Link selection generates a FROM clause. See the EQL Command Language.

The following keys are available in this option:

Enter To select a File or Link or schema name.

F2 To delete the Link or Schema selection. If you delete a Link (or Schema), the Query Framer automatically removes all references to columns or data in that Link from the query.

F4 To return to the Query Control Window.

F10 To lookup a specific file. The system prompts:

```
Enter Key Value
```

Type from 1 to 8 characters of the Link or Schema name and press **Enter**. The system displays the first file or Link name to match your entry.

Columns

This option selects the Data Names from the file (or files) selected to be used in this query.

The system displays a listing of Data Names. In Query-IV Preferences you may define the order in which they display as either alphabetical or physical

> You must select at least one column before executing the query.

The length of the Data Name is also displayed and can be used to determine whether the Data Name is numeric or alphanumeric.

Examples:

3.0 8.2 1.0 4.2 6.2
Represents numeric lengths. The field size depends upon the numeric type defined in the data file format. For more information, see the Formats section of the Dictionary-IV Developer Guide.

58 1 30 15
Represents alphanumeric lengths or 58, 1, 30, and 15) as defined in the data file format. For more information, see the Formats section of Dictionary-IV Developer Guide.

The selected column is then displayed at the bottom of the screen in the selected column area. The selected column area displays the following information for all selected columns:

- Column #
- Column Heading
- Data Name
- Numeric Output Mask

If you select a column that is defined as a multiple occurrence the system prompts:

```
Which Occurrence (1 thru x)
```

`x` is the number of occurrences. Type the occurrence to be used and press **Enter**. You may continue to add the other occurrences.

The following keys are available from this option:

Enter To select a column.

Left Arrow To move one column to the left in the display of selected columns at the bottom of the screen.

Right Arrow To move one column to the right in the display of selected columns at the bottom of the screen.

Tab To move one window width to the right in the display of selected columns at the bottom of the screen.

Back Tab To move one window width to the left in the display of selected columns at the bottom of the screen.

Home To move to the first (or last) Data Name in the listing.

Up Arrow To move one line up in the Data Name listing.

Down Arrow To move one line down in the Data Name listing**.**

F2 To delete the highlighted column from the selected column area.

F3 To reorder the columns. For more information, see the **F7** Other Options section later in this section.

F4 To return to the Query Control Window.

F7 To open the Other Options Window. These options allow you to place headings on the columns, add and delete columns, and reorder columns. As you make your selections, the columns displayed at the bottom of the screen will change. For more information, see the **F7** Other Options section later in this section.

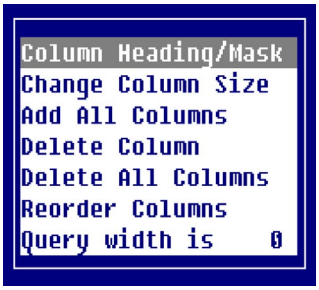
F10 To lookup a specific file. The system prompts:

```
Enter Key Value
```

Type from 1 to 20 characters for the Data Name and press **Enter**. The system places the cursor at the first matching Data Name.

F7 other options

When you press **F7** the system displays:



You may select one of the following:

Column Heading/Mask
This option allows the creation of headings for a column. The default heading will be the same name found in the Data Name list window. If the column contains numeric data, an optional output mask can be defined.

When you select this option and press **Enter**, the system prompts:

```
Column Heading:
Column Mask:
```

The default heading is the Data Name (or description). The field will scroll to accept a maximum of 60 characters, but only the first 20 characters are displayed in the selected column area, even though the entire heading will be used in the query.

> You can use a pipe (|) character in the column heading to specify a line break in the heading. The heading text must be fully enclosed in quotation marks.

If you select a numeric Data Name or a phone number, date, or social security number field, the system prompts for a numeric output mask.

Press **Enter** to accept the default, or enter another mask. For more information on masking, see the EQL Command Language section later in this document.

Change Column Size
This selection allows you to change the size of any character data column. The column name being changed is highlighted to the left. The new column size cannot be greater than the current column size.

When you select this option and press **Enter**, the system prompts:

```
Enter column size: x
```

x is the maximum field length. Type the column size to be used in this query and press **Enter**. This number must be equal to or less than the maximum field length.

> Only alphanumeric Data Names may have their columns resized.

If the column is a multiple occurrence the system prompts:

```
Which Occurrence (1 thru x)
```

x is the number of occurrences. Type the occurrence to be changed and press **Enter**. You may continue to change the size of other occurrences.

When you exit this option the system adds a line to the selected column area:

Column Size
This is represented by:

x represents the new column size.

Add All Columns
When you select this option and press **Enter**, the system adds all columns in the specified Format/Link to the query. If a Data Name specifies multiple occurrences, all occurrences will appear in the query.

The columns appear in the order in which they appear in the Data Name window. For more information about changing this order, see the Query-IV Preferences section in this document.

Delete Column
When you select this option and press **Enter**, the system deletes the selected column from the current position. If there are multiple occurrences the system prompts:

```
Which Occurrence (1 thru x)
```

x is the number of occurrences. Type the occurrence to be deleted and press **Enter**. You may continue to delete other occurrences.

Delete All Columns
When you select this option and press **Enter**, the system deletes all columns from the query. If a Data Name specifies multiple occurrences, all occurrences will be deleted from the query.

Reorder Columns
When you select this option and press **Enter**, the system displays the list of columns in your query.

The following keys are available:

Enter To select the column to move.

F4 To end.

Highlight a column and press **Enter** to select it. The system redisplays the list of columns in your query. The following keys are available:

Enter To select the new column position.

F4 To end.

> The column you selected to move will be positioned in front of the column you now select.

You may select additional columns to move or press **F4** to return to the column selection window.

> Reorder columns cannot move a column to the end of the list, but this can be accomplished by deleting the column and then reselecting it.

Query width is
The system displays the width, in number of characters, for a single row of data in the query you are building. This information will be used for screen scrolling and condensed printer printing.

The report width is rounded up to 80, 96, or 132.

Column Totals

This option selects the numeric Data Names for which you want column totals.

The system displays a listing of selected numeric Data Names.

Any number of numeric Data Names may be selected for totaling. When a Data Name is selected, it is placed in the selected column totals window.

> "Column Totals" will generate a TOTAL clause in the EQL Command Language. See the EQL Command Language.

The following keys are available in this option.

Enter Selects the Data Name for totaling.

F2 Deletes the highlighted Data Name from the selected column totals window.

F8 Selects all numeric Data Names for totaling.

F9 Deletes all numeric Data Names from the selected column totals window.

Rows Where

This selection allows you to enter a condition for selecting rows.

The system displays an entry window. Type the selection conditions and press **Enter**. The condition consists of two or more values that interact with operators to form either a true or false result.

For more information about the operators and conditions, see the Command Language section in this document.

If no condition is specified, all rows are selected. The condition consists of two or more values that interact with operators to form either a true or false result.

The following keys are available from this option.

F4 Returns to the Query Control Window.

F6 Displays on-line help.

F10 Displays a list of Data Names from all selected Links or Schemas. The system displays a listing of Data Names.

F11 Displays the operand menu.

Example:

```
CUSTOMER-BALANCE > 1000
```

This selects rows where the customer balance is greater than 1000. If you enter an invalid condition, you cannot exit the window without correcting the problem or deleting the condition from the window.

The syntax for the condition in this window is described in the section on the EQL Command Language. In this entry window you do not have to enter the WHERE that precedes the condition.

> "Row Selection" will generate a WHERE clause in EQL Command Language. See the EQL Command Language.

Order Rows By

This option allows you to select a sort order.

The system displays a Sort selection window containing a list of existing sorts defined for the first Link or Schema. You may also create a Sort. It may look like the following:

```
SORT0      CUST-CODE
SORT1      CUST-NAME
. . .
Create a Sort
```

SORT0-SORTn
SORT0 through SORTn represents the sorts defined in Dictionary-IV. For more information, see the Links section of the Dictionary-IV Developer Guide.

> SORT0 is always sorted by the primary key.

The following keys are available:

Enter To select the Sort.

F4 To end.

> You may use EQL to create temporary sorts. For more information, see the EQL section in this document. You may define additional, permanent sorts in Thoroughbred Dictionary-IV. For more information, see the Links chapter of the Dictionary-IV Developer Guide.

Only one Sort may be selected at any given time. After you select a Sort, you automatically return to the Query Control Window.

You cannot use Order Rows By and Group Rows By in the same query. Grouping automatically orders rows by the selected group and overrides any sorting selected in the Order Rows By window.

> "Sort Selection" will generate an ORDER BY clause in EQL Command Language. See the EQL Command Language.

Create a Sort Order
This option allows the user to create a Sort order that is not an IDOL-IV predefined Sort. Query-IV will sort and order the results using this value. The user will be given the opportunity to select a single Data Name from the listing in ascending or descending order.

Group Rows By

This option allows you to group rows together by the specified Data Name and produces a summary row for each group.

The system displays a listing of Data Names.

Only one Data Name may be selected for grouping. When a Data Name is selected, it is placed in a window.

The following keys are available:

Enter To select a Data Name for grouping.

F2 To delete the highlighted Data Name from the Group Rows By window.

F4 To end.

Grouping can be used to display the unique values in an alphanumeric column. Compare the first query below, which does not use grouping, with the second query, which does.

Example:

Files (Links): Q4EMPLOY
Columns: DEPT-CODE
Rows Where: EMPLOYEE-NAME BETWEEN "A" AND "E"

Table Displayed:

Department Code
01
09
05
02
03
07
01

Notice that the code 01 is displayed twice.

Example:

Files (Links): Q4EMPLOY
Columns: DEPT-CODE
Rows Where: EMPLOYEE-NAME BETWEEN "A" AND "E"
Group Rows By: DEPT-CODE

Table Displayed:

Department Code
01
02
03
05
07
09

Notice that the code 01 is displayed only once.

If a numeric column is selected, the query displays a single number for each group based on the group function used in the query. For the SUM function, it is the sum of the values. Compare the first query below, which does not use grouping, with the second query, which does.

Example:

Files (Links): Q4EMPLOY
Columns: DEPT-CODE, BONUS
Column Totals: BONUS
Rows Where: EMPLOYEE-NAME BETWEEN "A" AND "E"

Table Displayed:

Department Code	Bonus
01	1500
09	2800
05	18000
02	1500
03	6750
07	19500
01	1850
	51900

Again notice that department 01 displays twice, once for each employee.

Example:

Files (Links): Q4EMPLOY
Columns: DEPT-CODE, BONUS
Column Totals: BONUS
Rows Where: EMPLOYEE-NAME BETWEEN "A" AND "E"
Group Rows By: DEPT-CODE

When you execute this query the system prompts:

```
Use GROUP BY as Summary or Detail
```

Select one of the following:

Summary
Displays the Group By Data Name (DEPT-CODE) and totals the numeric columns selected (BONUS).

Detail
Displays all detail for each group.

Summary Table Displayed:

Department Code	Bonus
01	3350
02	1500
03	6750
05	18000
07	19500
09	2800
	51900

Notice that each department displays only once showing the bonus for each department and a total for all departments that fit the criteria.

Detail Table Displayed:

Department Code	Bonus
01	1500
01	1850
	3350
03	6750
	6750
05	18000
	18000
07	19500
	19500
09	2800
	2800
	51900

Notice that each employee bonus is displayed for each department; each department is totaled; and all departments fitting the criteria are totaled.

You cannot use Group Rows By and Order Rows By in the same query. Grouping automatically orders rows by the selected group and overrides any sorting selected in the Order Rows By window.

When a group is selected, the only columns that can be selected in your query are those that use a group function or are specified as a group in Group Rows By. This is an SQL requirement and will produce an error message if violated. In Thoroughbred Query-IV, the Query Framer helps you by automatically framing your query command to meet these requirements (rather than by giving you an error message like SQL).

If you select a group, the Query Framer automatically uses SUM for all selected numeric columns and ignores all alphanumeric columns except those specified in the group.

In EQL Group selection generates a GROUP BY clause. See the EQL Command Language.

Execute (Framer)

This option executes the query framed by the Query Framer and produces a table of information from the request.

It also generates and displays an SQL statement on the screen.

Summary
For the last example on the previous page a Summary query would produce:

```
TOTAL BONUS;
SELECT DEPT-CODE, SUM(BONUS)
FROM Q4EMPLOY
WHERE CUST-NAME BETWEEN "A" AND "E"
GROUP BY REP-CODE
```

Detail
For the last example on the previous page a Detail query would produce:

```
DETAIL
TOTAL BONUS;
SELECT DEPT-CODE, SUM(BONUS)
FROM Q4EMPLOY
WHERE EMPLOYEE-NAME BETWEEN "A" AND "E"
GROUP BY DEPT-CODE
```

For more details, refer to the section on Executing a Query.

Table scrolling

When you execute a query and the table displays on the screen, you may need to scroll the table to display all the columns and rows.

The following keys are available:

Enter To display additional rows.

Home To move to the left/right side of the table.

Right Arrow To move the window right 1 character.

Left Arrow To move the window left 1 character.

Tab To move the window right 5 characters.

Back Tab To move the window left 5 characters.

EQL COMMAND LANGUAGE

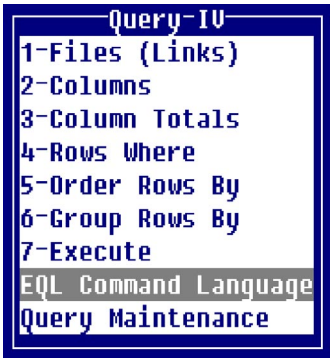
The EQL Command Language is an SQL-compatible query language. The EQL Command Language option in the Query Control Window displays an interactive window where you can directly enter and edit a query. It provides the full power and capabilities of Thoroughbred Query-IV. For more information about SQL, see the Introduction in this document.

You can generate a query in this window from the selections made using the Query Framer. You can also execute a query directly from this window.

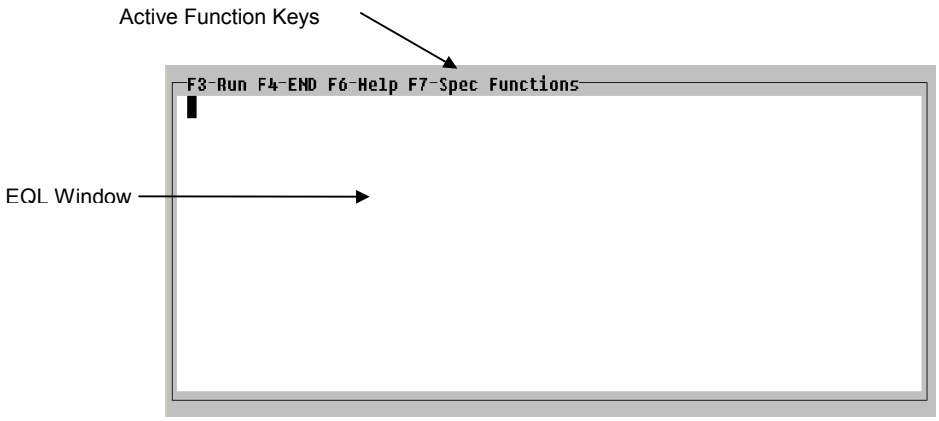
Executing the EQL window

When you run Thoroughbred Query-IV the system displays the Query Control window.

Query Control Window – EQL



Select EQL Command Language and the system displays the EQL window:



You may begin typing commands.

The following keys are available:

F3 To execute the command.

F4 To end.

F6 To view Help.

F7 To view and select from the special functions listing.

F7 - Special Functions

The following options are available from Special Functions:

Split Text
Splits the text. For more information, see the Introduction in this document.

Join Text
Joins the text. For more information, see the Introduction in this document.

Execute Command
Executes the command in the window.

End
Exits.

Display Sorts
Displays the sorts for the Link or Schema defined in your command. You may make selections from this listing.

Help
Displays on-line help.

Display Links
Displays a list of Links as defined in Operator Preferences. (EQL only works with links not schemas.) For more information, see the Operator Preferences section of this document.

Generate Command
Displays the EQL command for a query created in the Query Framer. For more information, see the EQL Command section in the Using Query-IV section of this document.

Display Data Names
Displays a list of Data Names for the Link or Schema defined in your command. You may make selections from this listing.

Operator Preferences
Displays Query-IV Operator Preferences to view or edit. For more information, see the Operator Preferences section of this document.

EQL syntax conventions

The following conventions are used to describe syntax in Thoroughbred Query-IV. For more information, see the Introduction of this document.

Notational
For the syntax notational conventions see the Introduction of this document.

String Data
Alphanumeric, or string, characters are identified by delimiting the data using double quotation marks. Although single quotation marks can be used, it is recommended that you standardize on using double quotation marks.

Example:

```
WHERE NAME = "TRANE"
```

Numeric data

Valid numeric data include digits from 0 to 9 and the following characters (commas are not valid):

```
-       +       .       E
```

Examples:

```
3490.24-25701   .1E5       .23E-5     32.85
```

Repeating Data Names (multiple occurrences)

A repeating Data Name has multiple occurrences. The number of occurrences is specified when the Data Name is defined. A repeating Data Name is really a table of Data Names that share a common name and definition.

To use a repeating Data Name, you must specify the Data Name and an offset, or occurrence number. If the offset number is not specified, the first occurrence is used as the default.

Substrings cannot be used with a repeating Data Name.

Do not use a space between the data name and occurrence specification or in the occurrence specification.

Example:

```
SELECT ITEM-CODE(3) FROM INVENTRY
```

This specifies the third occurrence of a multiple occurrence group.

Substrings

Substrings can be used with alphanumeric Data Names to specify a portion of the original value.

Substrings can only be used with alphanumeric Data Names.

Substrings cannot be used with a repeating Data Name (multiple occurrence).

The syntax is as follows:

```
string-elmt[(start,length)]
```

string-elmt	specifies an alphanumeric Data Name.
start	specifies the starting position for the substring.
length	specifies the number of characters of the substring.

Example:

```
SELECT CUST-NAME(1,5) FROM CUSMSTR
```

This specifies the first 5 characters of the customer name.

Masks (numeric output formats)

A mask specifies the format in which numeric data are displayed or printed.

Masks allow you to specify the same output format regardless of the size of the number. They also allow you to output financial symbols and characters (for example, $5,632.04-), and to retain insignificant zeros (e.g., output 1 as 1.00).

The syntax is as follows:

```
num-elmt:"mask"
```

num-elmt	specifies a numeric Data Name.
mask	specifies 1 or more mask characters.

Example:

```
SELECT BONUS:"$#,###.00",HRS-WORK*RATE:
"$#,##0.00"FROM PAYROLL
```

This specifies the placement of a dollar sign, commas, and decimal place in the designated Data Names.

Mask characters

For more information, see the Introduction of this document.

Special operators in conditions

NOT value
Negative condition. The NOT operator is used to negate a condition. It can be used to negate the result of the IN, BETWEEN, or LIKE operators.

IN (constant [, constant] ...)
Matching value in list. Items in the list specified by the IN operator must each be separated by a comma and space. Alphanumeric values must be padded to the correct length for a proper match.

LIKE "partial-value"
Partial equality. The LIKE operator can specify a string value containing wildcards, which can match more than 1 character. LIKE automatically pads its values to the correct length. (You must manually pad values compared by other operators.)

LIKE Wildcards

*	Matches any characters (0 or more).
?	Matches a single character.
[A-Z]	Matches a range for a single character.
[AGCF]	Matches a single character in list.
[wildcard]	Matches the wildcard character.

The brackets [ ] in wildcards are required; they do not indicate optional values. The * and ? wildcards do case-insensitive comparisons.

BETWEEN low-value AND hi-value
Greater than or equal to and less than or equal to. This operator can work with numeric or alphanumeric values, but the low-value and hi-value must be of the same type. Alphanumeric values must be padded to the correct length if you want to obtain a specific match for starting and ending values.

RANGE FROM "low-value" TO "hi-value"
Primary key field greater than or equal to and less than or equal to. This operator works only with alphanumeric values. It uses the primary key field from the first Link defined in the FROM clause to select a range of records. The key field does not have to be used as a column in the query. Alphanumeric values must be padded to the correct length if you want to obtain a specific match for starting and ending records.

SOUNDEX value
Sounds like. The SOUNDEX operator will compare values using phonetic operation to find data that has a similar pronunciation.

```
[output-options;]
SELECT column[,column]...
FROM link-name[,link-name]...
   [WHERE condition]
   [GROUP BY data-name]
   [ORDER BY sort name|temp-sort-definition];
```

output-options
    [DETAIL]
    [HEADING]
    [INPUT prompt INTO local data-name]
    [PRINTER[printer-id]]
    [TOTAL column[,column]...]

column
    value["column-heading"][:"mask"]
        value:
            constant,data name,expression,expression-alias, or group function
                expression-alias:
                    expression AS name
                group-function:
                    SUM(numeric-data-name|numeric-expression)
                    AVG(numeric-data-name)
                    MIN(numeric-data-name)
                    MAX(numeric-data-name)
                    COUNT(data-name)

condition
    Two or more values, consisting of constants, data names, or expressions that interact with relational or logical operators to form either a true or false result.

sort-name
    SORTx

temp-sort-definition
    data-name[([start-pos,length]|[A|D],[C])][+data-name…]…

[output-options]

The output options allow you to format your query to resemble a report, with a report title (query heading), grand totals on numeric columns, and the ability to obtain a printed copy of the query.

The output options can appear in any order, but as a group, they must be terminated by a single semicolon.

Example:

```
TOTAL CUS-BAL
HEADING
PRINTER;
SELECT CUS-CODE, CUS-NAME
FROM CUSMSTR
```

[DETAIL]

Allows you to display the detail lines that make up a summary row described by the GROUP BY clause.

Example:

```
DETAIL;
SELECT EMPLOYEE-NAME, DEPT-CODE, SUM(SALARY)
FROM Q4EMPLOY
GROUP BY DEPT-CODE
```

[HEADING]

This option specifies that a standard table heading, or title, is to be produced at the top of the query table. The standard heading has 3 lines, which contain the following information:

1. Date, Installation Name, Page Number
2. Query Description or Default Description
3. Blank Line

Example:

```
HEADING;
SELECT CUS-CODE, CUS-NAME, SALES-REP, YTD-SALES
FROM CUSMSTR
```

Table Displayed:

```
10/28/... Thoroughbred Software International, Inc. ...
Page  1        Customer YTD-Sales

CUS-CODE    CUS-NAME    SALES-REP   YTD-SALES
C100        Abe Stiltz  S3          11431
C200        Rita Mole   S2          43020
```

[INPUT prompt INTO local data-name]

Allows the creation of run-time retrieval rules. Prompts may be entered into your queries.

Example:

```
INPUT "Enter Dept:" INTO TMP-DEPT(2);
SELECT EMPLOYEE-NAME, DEPT-CODE
FROM Q4EMPLOY
WHERE DEPT-CODE=TMP-DEPT
```

[PRINTER[printer-id]]

All queries display on the screen by default, unless the "PRINTER" option is used. This option is ignored when the query is executed from a Script or Menu.

If a printer ID is specified, that printer is where the query will be output. If PRINTER is specified with no printer ID, the system asks you to select a printer when the query is executed.

Example:

```
PRINTER P1;
SELECT CUS-CODE, CUS-NAME
FROM CUSMSTR
```

[TOTAL column[,column]...]

This option allows you to obtain grand totals for any numeric column in your query. The TOTAL column definition must be a numeric Data Name or a numeric expression-alias name defined in the selected columns. No column heading or mask can be specified by the TOTAL. The total is displayed using the same masking specified for the column. If you specify multiple columns, you must separate each of them by a comma and space.

The order of the TOTAL columns is insignificant.

The grand total is produced at the end of the query under the appropriate column. The grand total consists of 2 lines: a line of dashes and the grand total line.

See SELECT column

```
TOTAL YTD-SALES, MARGIN;
SELECT CUS-CODE, CUS-NAME, YTD-SALES,
          YTD-SALES - YTD-COST AS MARGIN
FROM CUSMSTR
```

SELECT column[,column]...

SELECT is the command used to examine data in the database and is the primary command of query languages. You are required to specify SELECT in your query with at least one column. If you specify multiple columns, you must separate each of them by a comma and space.

column

A column can be defined as a constant, Data Name, expression, expression-alias, or group function. Once defined you can then specify a column heading and mask.

```
value["col-head"][:"mask"]
```

value is a constant, Data Name, expression, expression-alias, or group function.

col-head [optional] is the column heading to appear on the query.

mask [optional] is the masking to be used for the data in this column.

A column can specify either an alphanumeric (string) value or a numeric value.

All columns can specify a column heading. If a heading is not specified, the query uses the column definition as the heading. In the case of an expression, the entire expression is used, unless an expression alias is specified.

Only numeric columns can specify a mask. If you specify a column heading or mask, it must appear after the value and must be separated from surrounding items by a space. Each column definition is separated by a comma and space.

Example:

```
SELECT CUS-NAME, CUS-BALANCE
FROM CUSMSTR
```

You can include single quotation marks in a column heading, but you must use the single quotes *inside* the double quotes that delimit the heading.

> You can use a pipe, or vertical bar, character in the column heading to specify a line break in the heading. The heading text must be fully enclosed in quotes.

Examples:

```
SELECT CUS-NAME "Customer", CUS-PHONE
FROM CUSMSTR

SELECT EMP-LAST-NAME + ", " + EMP-FIRST-NAME "Name", EMP-GROSS-WAGES
:"$###,###.00"
         "Year-To-Date|Gross Wages"
FROM EMPMSTR
```

Data names
Alphanumeric data names can specify subscripts.

Example:

```
SELECT DESC(1,5)
FROM INVENTRY
```

If Data Name definitions include a description, the description will be used as the column heading.

Repeating Data Names can specify an offset, or occurrence number.

Example:

```
SELECT MONTH(5)
FROM CALENDAR
```

expression
Two or more elements, consisting of constants or Data Names that interact with operators to form a new value.

The field length for an alphanumeric expression is the combined length of all the values in the expression.

The field length used for a numeric expression is the largest field length used by a Data Name in the expression. A numeric expression can be followed by a mask. (It is recommended that you use a mask with numeric expressions.)

Example:

```
SELECT EMP-HOURS-OT * EMP-OT-RATE :"$#,###.00",
      EMP-LAST-NAME + ", " + EMP-FIRST-NAME(1,10)
FROM EMPMSTR
```

expression-alias
This option allows you to use a name to refer to a column. This is useful when the column definition is an expression that you are going to use elsewhere in the statement. By referring to the expression alias, you can save time and avoid making any keyboard entry errors when duplicating the expression. Also, the column heading uses the alias name as a default, rather than the entire expression.

The expression-alias name can be used anywhere that an expression can be used. It retains its original meaning. It cannot be used as a Data Name (a substring of an expression-alias name is invalid). The expression-alias name can also be used in some places where an expression cannot be used, such as in the TOTAL clause, or on the left side of an operator in the WHERE clause.

Example:

```
SELECT OPEN-INVOICE-AMT-ON-ACCOUNT-AMT AS BALANCE FROM CUSMSTR
WHERE BALANCE >500
```

group-function
This function finds the sum (SUM), average (AVG), minimum value (MIN), maximum value (MAX), and total number (COUNT) of all values in a numeric column and displays a single value.

The GROUP BY clause is used to control the level at which group functions are computed.

Do not use a space between the function name and the parenthesis or between the parenthesis and the Data Name or expression.

Query languages require the query to return the same number of values for each column selected. For this reason, group functions cannot be mixed with columns that return multiple values, and group functions cannot be used in the WHERE clause.

SUM

SUM(num-data-name|num-expression)

num-data-name is the numeric Data Name to be totaled.

num-expression is the numeric expression (addition, subtraction, multiplication, division, exponentiation) of Data Names before totaling.

Example:

```
SELECT AVG(SALARY)
FROM EMPLOYEE
```

This displays a single-row query containing a total of all employee salaries.

AVERAGE

```
AVG(num-data-name)
```

num-data-name is the numeric Data Name to be averaged.

Example:

```
SELECT AVG(SALARY)
FROM EMPLOYEE
```

This displays a single-row query containing the average of all employee salaries.

MINIMUM

```
MIN(num-data-name)
```

num-data-name is the numeric Data Name to check for and display the minimum value.

Example:

```
SELECT MIN(SALARY)
FROM EMPLOYEE
```

This displays a single-row query containing the minimum value of all employee salaries.

MAXIMUM

```
MAX(num-data-name)
```

num-data-name is the numeric Data Name to check for and display the maximum value.

Example:

```
SELECT MAX(SALARY)
FROM EMPLOYEE
```

This displays a single-row query containing the maximum value of all employee salaries.

COUNT

```
COUNT(data-name)
```

data-name is the Data Name to be counted.

Example:

```
SELECT COUNT(EMPLOYEE-NAME)
FROM EMPLOYEE
```

This displays a single-row query containing the total number of employee names.

FROM link-name [,link-name]...

The FROM clause specifies the Links (data files) from which you wish to examine data in your database. In your SELECT command, you are required to specify a FROM clause with at least one Link name.

If you specify multiple Link names, they must be related to each other by a common Data Name, and in the FROM clause, you must separate each of them by a comma and space.

The names of all Links that have any of their columns referenced anywhere in the query must appear in the FROM clause.

Data Names from all specified Links are available to be used as a selected column. If the spelling of a Data Name in one selected Link matches a Data Name in another selected Link, the Data Names are treated as one name. Like Data Names should contain like data.

Example:

```
SELECT SLSPSN-NAME, CUS-NAME, INVOICE-NUM,
      INVOICE-AMT
FROM SLSPMSTR, CUSMSTR, INVMSTR
```

[WHERE condition]

The WHERE clause is optional and is used to select rows for the query based upon a specified condition. If this clause is omitted, all rows are selected.

condition is two or more values, consisting of constants, Data Names, or expressions that interact with relational or logical operators to form either a true or false result.

Values or columns used in the WHERE clause need not be selected by the query.

An expression can only be specified on the right side of an operator, but an expression-alias name can be specified on either the right or left side.

The NOT operator can be used to negate the result of the IN, BETWEEN, or LIKE operators.

The brackets [ ] in wildcards are required; they do not indicate optional values.

The LIKE operator can specify a string value containing wildcards, which can match more than 1 character. LIKE automatically pads its values to the correct length. You must manually pad values compared by other operators.

Comparisons are case-sensitive with the exception of the * and ? wildcards.

Items in the list specified by the IN operator must each be separated by a comma and space. Using the IN operator is equivalent to specifying multiple equality tests (=) grouped by OR. Using NOT IN is equivalent to specifying inequality tests (< >) grouped by AND.

Group functions, such as SUM, cannot be used in the WHERE condition.

Examples:

```
SELECT ITEM, ITEM-DESC
FROM INVENTRY
WHERE ITEM LIKE "M*M"
```

Selects rows where the item starts with an M and ends with an M.

```
WHERE ITEM NOT LIKE "C*"
```

Selects rows where the item starts with anything but a C.

```
WHERE ITEM LIKE "[A-E]900*"
```

Selects rows where the item starts with an A, B, C, D, or E followed by 900.

```
WHERE ITEM IN("X7329M", "X7020A", "Q9903L")
```

Selects rows where the item matches any of the three items in the list.

```
WHERE ITEM BETWEEN "D3195M" AND "E9999Z"
```

Selects rows where the item matches D3195M, E9999Z, or any item in between these two values.

```
WHERE ITEM NOT BETWEEN "D3195M" AND "E9999Z"
```

Selects rows where the item does not match D3195M, E999Z, and does not match any item in between these two values.

```
SELECT CUS-NAME, BALANCE
FROM CUSMSTR
WHERE RANGE FROM "C100" TO "C199"
```

Selects rows where the customer code matches C100, C199, or any customer code in between these two values.

```
SELECT CUS-NAME, BALANCE, CRED-STATUS
FROM CUSMSTR
WHERE BALANCE > 500 AND CRED-STATUS LIKE "SHAKY*"
```

Selects rows where the credit status matches "SHAKY". Using the LIKE operator allows you to specify a match without padding the value.

```
SELECT CUS-NAME, BALANCE, CRED-STATUS
FROM CUSMSTR
WHERE BALANCE > 500 AND CRED-STATUS = "SHAKY "
```

Selects the same rows as the previous example, but because the = operator was used instead of the LIKE operator, the value "SHAKY" had to be padded with spaces up to the defined length of CRED-STATUS to get the correct matches.

[GROUP BY data-name]

The GROUP BY clause is optional and is used to group rows together by the specified Data Name. It produces a summary row for each group.

`data-name` is the Data Name you want to group by.

GROUP BY controls the level at which the group function is computed. If a group function is specified as a selected column, the group function is applied to the group defined here. The group function operates on the group of records and produces a single value for each group.

This clause automatically orders the query by the specified group. The GROUP BY clause cannot be used when an ORDER BY clause is used; they are mutually exclusive.

If the specified group exists as a Thoroughbred IDOL-IV predefined Sort, or as the first part of a Thoroughbred IDOL-IV predefined Sort, that Sort is used to order the query (GROUP BY DEPT-CODE will use SORT1 DEPT-CODE + EMPLOYEE-NAME). This is done for the purpose of getting a quicker response to the query.

> When a group is selected, the only columns that should be selected in your query are those that use a group function or are specified as a group in the GROUP BY clause. This is a requirement of SQL.

Example:

```
SELECT DEPT-NUM
FROM EMPLOYEE
GROUP BY DEPT-NUM
This displays a summary that lists each unique department from the employee
file.

SELECT DEPT-NUM, SUM(SALARY)
FROM EMPLOYEE
GROUP BY DEPT-NUM
```

This displays a summary query that lists a total of employee salaries for each department.

[ORDER BY sort-name|temp-sort-def]

The ORDER BY clause is optional and is used to produce the query with the rows in a specified order. If this clause is omitted, the query is ordered by the primary key sequence of the first Link (SORT0).

The ORDER BY clause cannot be used when a GROUP BY clause is used; they are mutually exclusive. If a GROUP BY clause is used, the rows are automatically ordered by the specified group.

sort-name is a standard IDOL-IV sort name (SORTx). See below.

temp-sort-def is a standard IDOL-IV temporary sort file. See below.

SORTx
SORTx represents a predefined Sort. The x is a sort number from 1 to 99. SORT0 represents a sort by the primary key. For more information, see the Links chapter of the Dictionary-IV Developer Guide.

```
SELECT CUS-NAME, BALANCE
FROM CUSMSTR
ORDER BY SORT2
```

temp-sort-definition
This is a standard Thoroughbred IDOL-IV Sort definition that is used to order the rows in the query. The Sort definition creates a temporary Sort file that is used for the purposes of this query only.

This may delay your query when the data file is very large, in which case it is recommended that you use a Thoroughbred IDOL-IV Sort name rather than a temporary Sort definition.

```
data-name[([start-pos,length]|[A|D],[C])]
[+ data-name...]...
```

data-name is the Data Name by which the Sort is performed.

start-pos is the position to begin the substring. (Cannot be used with D or C options.)

len is the number of characters in the substring. (Cannot be used with D or C options.)

A (optional) is ascending order.

D (optional) is descending order.

C (optional) is case-sensitive. The default is case-insensitive.

A substring can be specified with the (start-pos,length) option. This option cannot be used with the A, D, or C options.

The A and D options are used to specify the sort order: ascending or descending. These options are mutually exclusive. If no sort order is specified, the order defaults to ascending.

The A option specifies that the sort order is ascending. The D option specifies that the sort order is descending.

The C option specifies the sorting to be case-sensitive. The default is case-insensitive. (Capital letters are uppercase; small letters are lowercase.)

A comma is needed only if an option precedes. The A, D and C options affect only the preceding Data Name and not the entire secondary key. Additional Data Names can be concatenated to form a multi-part Sort definition.

Example:

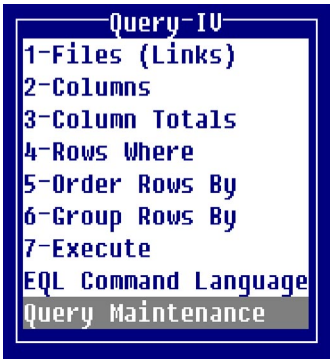
```
SELECT OPEN-INVOICE-AMT-ON-ACCOUNT-AMT AS BALANCE FROM CUSMSTR
ORDER BY BALANCE(D)
```

QUERY MAINTENANCE

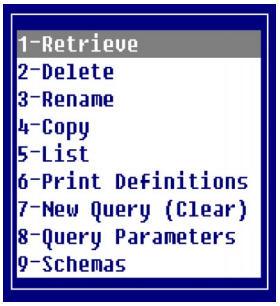
This chapter describes the options in the Query Maintenance Window, including how to start a new query, how to maintain existing queries (add, retrieve, delete, rename, copy and list) and how to print query definitions.

Executing query maintenance

When you run Thoroughbred Query-IV by pressing **F1** and selecting **Query** the system displays the Query View. Once a query is selected press **F1-Edit** and the system displays the Query Control Window:



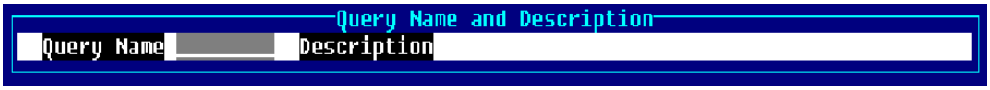
Press **Q** to highlight Query Maintenance, and press **Enter** to select it. The Query Maintenance Window contains the following options: retrieve, delete, rename, copy, list, print definitions, new query (clear), query parameters, and schemas.



Perform query maintenance

When you open Query-IV the workspace is empty and you may retrieve and create queries. Once you place a query in the workspace, it remains there until it is saved or cleared.

> If you have been working on a query and you select retrieve, the system prompts:
> **Save changes to current query?**
>
> You must save or clear the query currently in the workspace before you may retrieve a new query. You may also clear the query by selecting **7-New Query**.



Enter the following information:

Query Name
At the Query Name field, type a name from 3 to 8 characters long to identify the query and press **Enter**. (The first 2 characters are the library, used to group your queries together.)

If the library does not exist, you are allowed to create the new library automatically when the following prompt is displayed:

```
Creating new library: XX. Enter title:
```

If you did not intend to create a new library, press **F4**. Otherwise, enter an optional library title.

The following function keys are available at the Query Name field:

F2 Provides a listing of queries in a selection window.

F4 Cancels the selection and redisplays the Query Maintenance Window.

F6 Provides a window that lists existing queries and allows you to select one.

Description
Type up to 40 alphanumeric characters for an optional query description to identify the query. Press **Enter**. If the HEADING clause is specified in EQL, this query description is included in the table heading, overriding the default table heading.

1-Retrieve

This option allows you to place a query into the workspace.

Type a query name from 3 to 8 characters long and press **Enter**. When you select **1-Retrieve** and there is currently a query in the workspace, the system gives you the opportunity to save the current query.

When you retrieve a query, it becomes the current query. Press **F4** to return to the Query Control Window and you may edit this query. You may select any of the Query Framer options or the EQL Command Language to perform Query Maintenance.

When you have completed, press **F4** and the system prompts:

```
Save current query?
```

Select one of the following:

Y Saves the current query and prompts for a name description.

N Does not save the current query.

2-Delete

This option allows you to delete a query from the dictionary (where query definitions are saved).

Type a query name from 3 to 8 characters long and press **Enter**. The system prompts:

```
Are you sure you want to delete (Y/N)?
```

Select one of the following:

Y Deletes the query.

N Does not delete the query.

3-Rename

This option allows you to change the name of a query.

```
Enter new name:
```

When you rename the query, it does not have to remain in the same library. If you rename it and the library does not exist, you are allowed to create the library automatically.

4-Copy

This option allows you to copy a query, producing an identical copy of the query with a new name.

Type a query name from 3 to 8 characters long and press **Enter**. The system prompts:

```
Enter copy to name:
```

When you copy the query, it does not have to remain in the same library. If you copy it and the library does not exist, you can create the library automatically.

5-List

This option is used to print a list of queries.

Type a query name from 3 to 8 characters long and press **Enter**. The system prompts:

```
Select printer (xx xx):
```

Select the printer; xx represents a defined printer name for your system.

The query listing screen is displayed. Enter the following:

Library
Type the 2-character library name for which you want a query list.

Select everything?
Select one of the following:

Y All queries in the library will be listed. You cannot select a range or mask.

N Positions the cursor at the next field. You can select a range or mask.

from name
Type the 3 to 8 character name of the first query in the range and press **Enter**. The default is **First**. If a query name is entered at the from name field, press **Enter** at the to name field to enter the same name.

to name
Type the 3 to 8 character name of the last query in the range and press **Enter**. The default is **Last** (or the name is entered in the from name field).

using name mask
A name mask allows you to select queries without specifying an individual name. The default is **None**.

A mask sets up a matching test. If a query name matches the mask, then it is selected. If no match is made, it is bypassed.

A mask contains match and passing characters. The characters in a query name must match all of the match-characters (same character and position) for it to be selected. The pass character (?) is used to indicate that no match in this position is needed.

Mask	Action	Query Names		
XA?	selects	XAB	XAC	XA1
XA?	bypasses	XaB	XcA	X11

```
Is the displayed information correct?
```

Select one of the following:

Y Starts the listing.

N Allows you to change the information.

The listing is displayed or printed and includes library information, query name, description, query creation date, and last change date.

When the listing is finished, you are returned to the Query Maintenance Window.

6-Print Definitions

This option is used to print query and schema reports. The system displays the following:

Query and schema reports are copies of the actual query definitions, containing all of the specifications, content and format information that is recorded.

These reports are used primarily in design of the queries and as an aid for Query Maintenance. The following is a description of the prompts and responses.

The prompts for queries and schemas is the same.

```
Select printer (xx xx):
```

xx represents a defined printer name for your system.

The query reporting screen is displayed. Enter the following information:

Library
Type the 2 character library name for which you want to print.

Select everything?
Select one of the following:

Y All queries in the library will list. You cannot select a range or mask.

N Positions the cursor at the next field. You can select a range or mask.

from name
Type the 3 to 8 character name of the first query in the range and press **Enter**. The default is **First**. If a query name is entered at from name field, press **Enter** at the to name field to enter the same name.

to name
Type the 3 to 8 character name of the last query in the range and **Enter**. The default is **Last** (or the name is entered in the from name field).

using name mask
A name mask allows you to select queries without specifying an individual name. The default is **None**.

A mask sets up a matching test. If a query name matches the mask, then it is selected. If no match is made, it is bypassed.

A mask contains match and passing characters. The characters in a query name must match all of the match-characters (same character and position) for it to be selected. The pass character (?) is used to indicate that no match in this position is needed.

Mask	**Action**	**Query Names**		
XA?	selects	XAB	XAC	XA1
XA?	bypasses	XaB	XcA	X11

```
Is the displayed information correct?
```

Select one of the following:

Y Starts the printing.

N Allows you to change the information.

The system prompts:

```
Printing
```

The report is printed, including library information, query name, description, creation date, and last change date and time, as well as all the definitions from the Query Framer and EQL Command Window (e.g., Files/Links, columns, and column totals).

> If at any time you want to interrupt the printing, press the **Esc** key (on some systems, the **Del** key), and you will be asked if you want to interrupt the printing.

When the report is finished, you are returned to the Query Maintenance Window.

7-New Query (Clear)

Use this option if you have a query currently in the workspace and want to create a new query. You automatically start with an empty workspace when you run Query-IV.

When you select New Query, you are given the opportunity to save the current query (unless the current query was not changed):

```
Save current query?
```

Select one of the following:

Y Allows you to save the current query (requests a query name and description) before the query is cleared.

N Does not save the current query before the query is cleared.

The system prompts:

```
Query has been cleared, RETURN to continue.
```

After you clear a query, you return to the Query Maintenance Window.

8-Query Parameters

Directs you to the Operator Preferences option (**F7** Special Functions) in the Query Control Window. For more information, see the Operator Preferences chapter of this document.

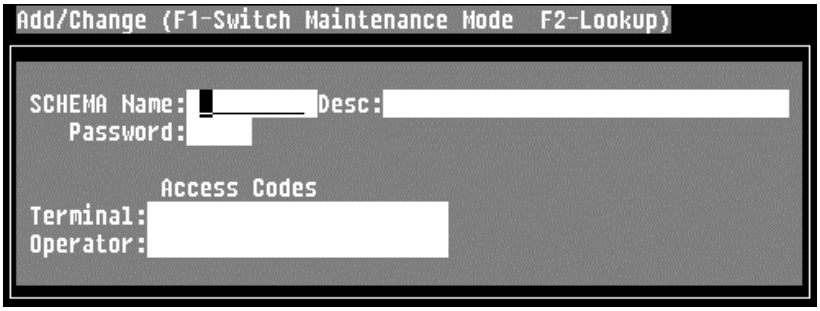
9-Schemas

This option allows you to create schemas.

A schema is a schematic diagram of a database that shows the relationships among various files. Often the data required to complete your query or request resides in multiple files. Query-IV allows you to select multiple files with just the data-names necessary to complete the schema.

You may use a schema in place of a Link during Query-IV file/table selection. A single schema definition can be used for multiple related files using selected Data Names from associated files, while the Link definition is designed for a single data file and format. For files to be related the key in one file and the Data Name in another file must be the same.

When you select this option the system displays the following screen:



SCHEMA Name
Type from 3 to 8 alphanumeric characters for the schema name and press **Enter**.

A schema may include multiple related Links and a subset of Data Names from those Links. When using schemas from the Query Framer Menu, you may access only one schema. Your data name selection will be limited to the names defined here in schema maintenance.

The following function keys are available:

F1 Switches maintenance mode (add/change, delete, rename, and copy).

F2 Displays a selection window of schema names.

F4 Exits.

F6 Displays on-line help.

Desc
Type up to 40 alphanumeric characters for the description of this schema and press **Enter**. This field is optional.

Password
Type from 1 to 3 alphanumeric characters for a password and press **Enter**. If the system security is active and a password is specified, the password is required. This field is optional.

Terminal Access
Use this field to allow access to or restrict access from the terminals.

```
0, term-code [,term-code]...
```

`0` Allows access to the specified terminal codes.

term-code Type the 2 alphanumeric character terminal codes for the terminals that can access this selection.

If this is left blank all terminals have access.

1, term-code [,term-code]...

1 Denies access to the specified terminal codes.

term-code Type the 2 alphanumeric character terminal codes for the terminals that cannot access this selection.

If this is left blank all terminals have access. Terminal access is not applicable to scripts.

Examples:

0,T0,T3
Allows access for terminals T0 and T3.

1,T1
Denies access for terminal T1.

Operator Access
Use this field to allow access to or **restrict access from the specified operators.**

0, oper-code [,oper-code]...

0 **Allows access to the specified operator codes.**

oper-code Type up to 3 alphanumeric characters for the codes of the operators that can access this selection.

If this is left blank all operators have access.

1,oper-code [,oper-code]...

1 Denies access to the specified operator codes.

oper-code Type up to 3 alphanumeric characters for the codes of the operators that cannot access this selection.

If this is left blank all operators have access. Operator access is applicable to scripts.

Examples:

0,JBT,NRS
Allows access for terminals JBT and NRS.

1,JMG
Denies access for terminal JMG.

The system displays the following:

Link Names
You must select at least one Link before selecting any other options in the Schema Definition Window.

The following keys are available:

Enter Selects the Link.

F2 Deletes the selected Link from the window.

F4 Exits.

F6 Displays on-line help.

The Link selection window displays a list of Links. More than one Link may be selected but the Links must be related. After a Link is selected, it is placed in the selected Links window and the list is redisplayed (starting at the selected Link).

Data Names
The Data Names selection window displays a list of Data Names from all selected Links. They display in alphabetical order. The system assumes duplicate Data Names are the same and display them only once.

Any number of Data Names may be selected. The selected Data Name is then displayed at the bottom of the screen in the selected Data Name area.

The following keys are available:

Enter Selects a Data Name and displays it in the Data Name area.

F2 Deletes the selected Data Name from the selected Data Name area.

F4 Exits.

F6 Displays on-line help.

F8 Selects all Data Names from the selected Data Name area.

F9 Deletes all Data Names from the selected Data Name area.

Sample schema:

A schema may look like the following:

```
SCHEMA REPORT
     Name:    Q4EMPING Employee Information Database
======================================


Files (Links):
     Q4EMPLOY, Q4DEPT

Available Data-Names:
     BONUS, DEPT-CODE, DEPT-LOCATION, DEPT-NAME,
EMPLOYEE-CODE, EMPLOYEE-NAME,
     JOB-DESCRIPTION, SALARY
```

QUERY EXECUTION

This chapter describes how to execute a query and display the table of information requested by the query. If some of the columns or rows in the table do not fit on the screen, you can scroll them onto the screen using window control.

Syntax Error
If you execute a query and get a syntax error, refer to the Messages chapter of this document.

Security
Query execution is restricted by Thoroughbred IDOL-IV system security. This includes Link and Schema security (restricted by password and terminal and operator access) and Format security (restricted data display for Data Name).

Selection methods

When you run Thoroughbred Query-IV the system displays the Query Control window.

If you access Thoroughbred Query-IV by pressing **F1** and selecting **Query** the system displays the Query View. From the Query View press **F1** to execute and the system will display the Query Control window.

You may also execute a query from a 3GL program or 4GL program (script), or from a Thoroughbred IDOL-IV menu.

From Thoroughbred Query-IV

Query Framer
If you have selected a query through the Query Framer, select **7-Execute (Framer)** in the Query Control Window to execute the query.

This selection is exclusively used to execute the Framer command. If there is no command in the Query Framer, nothing is executed, even if a command exists in the EQL window.

EQL
Use **F3** in the EQL window to execute the command in the window. If you want to execute directly from the EQL window the query selected in the Query Framer, use **F9**. This generates the EQL command from the selections in the Query Framer.

The command in the EQL window is executed. If no command is in the EQL window, the query is ignored.

IDOL-IV Menu
To execute a query from a Thoroughbred IDOL-IV menu, create a menu selection in the Menu definition. Use any selection code of your choice, and specify type □Q□. If you specify a query name as the action, it will automatically be executed. If no name is specified, the system will prompt for the query name to execute.

From a 4GL script

When you execute a query from a script or program, the following options are available:

```
Execute a range of queries in a library.
Display the output on screen or print a hard copy.
```

You must specify the options before you execute the query. A number of options are available. For more information, see CONNECT QUERY in the Script-IV Language Reference Manual.

The PRINTER option is ignored when you execute a query from a Menu or a Script.

Q4AUTO program

The sample Thoroughbred Basic program, Q4AUTO, which appears on the following pages demonstrates how to set up the parameters and execute the query from a program. A similar procedure can be used in scripts.

Program Q4AUTO (Page 1)

```
--------------------------------------------------------------------------
LIBRARY: Q4     TITLE: QUERY-IV Sample Ext. Interface   CREATED: 03/03/97
PROGRAM: Q4AUTO OWNER: TSI                            LAST CHANGED: 10/09/97
ALIAS NAMES:
--------------------------------------------------------------------------
10/09/97    Q4-Q4AUTO        QUERY-IV Sample Ext. Interface     PAGE: 1
--------------------------------------------------------------------------

  *   Interfacing Thoroughbred QUERY-IV with Other Application Programs
  *      This sample program illustrates how you can integrate queries
  *      into application scripts or programs.
  *      This allows you to embed Thoroughbred queries in your own
  *      systems.
  *
  0010   REM "Q4AUTO - Thoroughbred QUERY-IV Sample Interface"
  0020   BEGIN

  !   This first segment allows the operator to select the options.

  0100   REM "*** INPUT QUERY TO RUN ***"
  0110   INPUT 'CS',@(0,17),"Enter Library Name: ",L$;
         IF CTL=4
            GOTO 300
         ELSE
            IF LEN(L$)<>2
               GOTO 110
  0120   INPUT @(0,18),"Enter FROM 6 character query name: ",N$;
         IF CTL=3
            GOTO 110
         ELSE
            IF CTL=4
               GOTO 300
         ELSE
            IF LEN(N$)>6
               GOTO 120
            ELSE
               IF N$=""
                  N$=$000000000000$
```

Program Q4AUTO (Page 2)

```
--------------------------------------------------------------------------------

10/09/97     Q4-Q4AUTO        QUERY-IV Sample Ext. Interface    PAGE     2
--------------------------------------------------------------------------------

   0130   INPUT @(0,19),"Enter TO 6 character query name: ",N1$;
          IF CTL=3
             GOTO 120
          ELSE
             IF CTL=4
                GOTO 300
             ELSE
                IF LEN(N1$)>6
                   GOTO 130
                ELSE
                   IF N1$=""
                      N1$=$FFFFFFFFFFFF$
   0140   INPUT @(0,20),"Enter MASK 6 character query name:"N2$;
          IF CTL=3
             GOTO 130
          ELSE
             IF CTL=4
                GOTO 300
             ELSE
                IF LEN(N2$)>6
                   GOTO 140
   0160   INPUT (0,ERR=0160) @(0,21),'CL',
             "Do you want a hard copy? (Y?N) ",
             H$(""=0110,"Y"=0170,"N"=0170)

   !  The following is the routine that builds the values required to run
   !  the specific query. This code can be placed in any program.

   0200   REM"*** BUILD X$FOR THOROUGHBRED QUERY-IV ***"

   0210   DIM X$(84)           !  Required as shown.
   0220   X$(1,4)="AUTO"       !  Required as shown.
   0222   X$(84,1)="q"         !  Required as shown.

                               !  Starting Query Name:
   0230   X$(5,8)=L$+N$        !  First 2 characters are Library Name.
                               !  Characters 3-8 selects the starting query.

                               !  Ending Query Name (without Library).
   0240   X$(72,6)=N1$         !  Characters 1-6 are the ending query.
```

Program Q4AUTO (Page 3)

```
--------------------------------------------------------------------------
10/09/97       Q4-Q4AUTO    QUERY-IV Sample Ext. Interface       Page 3
--------------------------------------------------------------------------
  0250   X$(78,6)=N2$     !  Mask Query Name (without Library).
                           !  Characters 1-6 are the Mask Definition

  0260   X$(68,1)=H$       !  "N" = Screen Display
                           !  "Y" = Printed Query

  0270   X$(40,8)="ID"     !  This is the PROGRAM to Run after the Query
                           !  is completed. Set to any valid Program Name.

  0280   RUN "IRPCA0"      !  Executes the Query. Thoroughbred IDOL-IV
                           !  must be accessible.

  0300   RUN "ID"          !  Program to run if this program is terminated.
                           !  Set to any valid Program Name.
```

Table scrolling

When you execute a query and the query table is displayed on the screen, you may need to scroll the table to display all the columns and rows requested by the query. When the table is displayed, Thoroughbred Query-IV displays a prompt at the bottom of the screen, describing what you can do and allowing you to scroll the table.

Messages

`CR to continue`

There are more rows in the table than are displayed on the screen. Press **Enter** to display more rows.

`Enter Window Control`

There are more columns in the table than are displayed on the screen (the table is wider than the screen). You can scroll the table horizontally to display additional columns (see details below).

`End of Query`

There are no more rows; the last row of the table has been displayed.

F4 - END

Press **F4** to end the query and return to the place where you executed it.

Window control

The following keys are available at the **Enter Window Control** prompt:

Right Arrow To move the window right 1 character.

Left Arrow To move the window left 1 character.

Backspace To move the window left 1 character.

Up Arrow To move to the center of the table.

Home To move to the left and right sides of the table.

Tab To move the window right 5 characters.

Shift-Tab To move the window left 5 characters.

F9 To toggle between 80 and 132 character displays (where available).

MESSAGES

There are three types of messages that are displayed when you use Thoroughbred Query-IV:

- Syntax Error Messages
- Operational Messages
- System Error Messages

Syntax Error Messages
Syntax error messages are displayed when you execute an invalid query. In most cases, this can occur when you enter an invalid EQL command in the EQL Command Language Window. It can also occur when executing a command in the Query Framer, although this is less likely. These messages indicate that the EQL command cannot be executed and are self-explanatory, so that you can locate the problem and correct it yourself. Check the spelling of keywords, Data Names and Links. If you need more help, refer to the error message in this chapter for hints and possible causes. An error code precedes the syntax error message to identify it and help you locate the message in this chapter.

Operational Messages
Most operational messages are described in the appropriate chapters of this document. This chapter contains additional messages that may be displayed during the normal processing of your system and which are not explained in the body of the document, because in most cases they appear infrequently.

System Error Messages
If a system error occurs, please report it to the support department, along with all pertinent information (the software version, computer hardware, and details on the procedure you were performing when the error occurred).

Refer to the appropriate message section in this chapter to locate a specific message.

Syntax Error Messages

001 - Duplicate keyword XXXXX encountered.

The indicated keyword appears twice; one occurrence must be removed from the command.

002 - XXXXX is not a keyword.

The indicated word is not a keyword, although it is found in the position of a keyword. The keyword may be missing or you may have missing or extraneous punctuation in the command.

003 - Invalid link XXXXX.

The Link is not found or an error occurred while obtaining the Link. Use Thoroughbred Dictionary-IV to check the Link and fix the problem.

004 - Invalid format XXXXX.

The Format specified in the Link is not found or an error occurred while obtaining the Format. Use Thoroughbred Dictionary-IV to check the Link and Format and fix the problem.

005 - Unpaired parenthesis or quotes.

Starting punctuation was used without any ending punctuation to match. Fix the punctuation.

006 - Alias name XXXXX already defined.

The expression alias name specified by the AS keyword conflicts with an existing Data Name. Use a different name for the alias.

007 - Invalid expression.

The expression used for this column is invalid. The cause could be non-matching data types, incorrect use of arithmetic operators or invalid punctuation.

008 - Cannot find data name XXXXX.

The specified Data Name is not found in the Data Name list. The name is misspelled or the Link is not selected. Only Data Names from the selected links can be used in the query.

009 - Subscript value greater than number of occurrences.

The specified subscript is not a valid occurrence number, or offset number, for this repeating Data Name. Use Thoroughbred Dictionary-IV to check the Format definition for this Data Name and determine the defined number of occurrences.

010 - Substring exceeds data length.

You have specified a substring of an alphanumeric Data Name that exceeds the defined element length. Check the element length displayed in the Data Name list against your substring specification. The substring should specify a starting position and length that falls within the defined length of the Data Name.

011 - Invalid XXXXX.

The indicated item is invalid; usually this results from an invalid column heading or numeric output mask. Check for correct use of spacing, commas and quotes.

012 - ORDER BY not allowed with GROUP BY.

The ORDER BY clause cannot be used when a GROUP BY clause is used; they are mutually exclusive. (If a GROUP BY clause is used, the rows are automatically ordered by the specified group.)

013 - Match value in LIKE exceeds data length.

The match value specified by the LIKE operator exceeds the length of the column definition. Use another match value.

014 - Unable to recognize LIKE XXXXX.

The match value specified by the LIKE operator is invalid or unsupported. Use another match value.

015 - Invalid RANGE specification.

The specified range is invalid for this query. Possible causes include the following: (1) the FROM keyword or value is not found, or the TO keyword or value is not found, (2) the FROM and TO values cause a data type mismatch or do not specify a string value, or (3) the values cannot be used to access the file (the file is not a key-access file or the values do not specify the primary key of the first Link).

016 - Unmatching data types.

The query uses an alphanumeric value where a numeric value is expected, or a numeric value where an alphanumeric value is expected. This can easily occur in an expression, in a condition that compares values, or with the LIKE operator, which requires string values.

017 - No operators for XXXXX phrase.

A keyword appears without the required specifications. The keyword might be followed by the end of the query or by another keyword. Refer to the EQL syntax and add the required items.

018 - Unrecognizable syntax.

The syntax cannot be understood. This is commonly caused by a WHERE clause that is incomplete or invalid.

019 - SELECT and FROM are required for all queries.

The query does not contain a valid SELECT or FROM. You must select at least 1 column and specify at least 1 Link in the query in the Format: SELECT column FROM Link-name.

020 - Data elements must be from the same link.

The GROUP BY or ORDER BY clause specifies Data Names that are from more than 1 Link.

021 - Missing syntax element.

This is commonly caused by a WHERE condition that is incomplete.

999 - Thoroughbred Query-IV internal error.

Please contact product support for help with this problem.

***Syntax error, F1 to display, RETURN to continue.**

Please contact product support for help with this problem.

Operational messages

Cannot determine access to file 'XXXXX'.

This commonly occurs when 2 unrelated Links, or Files, are specified in the query. When more than 1 Link is specified in the query, it must be related to the preceding Link by a common field. It can also be caused by specifying data from Links other than the first Link in an ORDER BY or GROUP BY clause.

Cannot locate file XXXXX.

The data file specified in the Link does not exist or is inaccessible. Check with your system administrator.

Cannot locate IDOL-IV sort file.

The ORDER BY clause specifies an existing SORTx, but the Sort file cannot be located. Use Thoroughbred Dictionary-IV to check the Sort file name in the Link or to rebuild the Sort file.

Enter password for link 'XXXXX':

System security requires that you enter the Link password before the Link can be selected or the query executed.

File XXXXX busy, cannot open.

The specified file is unavailable at the moment. Try your query later or contact the system administrator.

Operator denied access to 'Link - XXXXX', RETURN to continue.

The Link contains operator security preventing the operator from accessing the Link. There is no way to bypass this restriction without removing it from the Link definition.

Security doesn't allow XXXXX to be selected, RETURN to continue.

The specified column (Data Name) contains Format security restricting the display of its contents. The Data Name cannot be selected and the query executed using this Data Name.

Sort key too large for file XXXXX.

The ORDER BY clause specifies a temporary Sort definition that exceeds the maximum length of 72 (on most systems).

Terminal denied access to 'Link - XXXXX', RETURN to continue.

The Link contains terminal security preventing the terminal from accessing the Link. There is no way to bypass this restriction without removing it from the Link definition.

XXXXX is not a valid file definition (format).

The Format specified in the Link definition cannot be located. Use Thoroughbred Dictionary-IV to check the Format name in the Link and also the Format definition to fix the problem.

System error messages

System errors are recognized by the display of a message on the top line of the screen:

```
System Error: Pgm-XXXXXXXX Ln-#### Err-## RETURN=Retry E=End
```

Pgm Indicates the BASIC program in which the error occurred.

Ln Indicates the line number at which the error occurred.

Err Indicates the BASIC error code.

Depending upon the error code and other factors, it may be possible to recover and continue. To continue, or retry, press **Enter**.

If the error recurs or cannot be corrected, press **E** to end the procedure, which takes you out of the process and tries to recover from the error as much as possible.

Common errors

Error 0 Record Busy or File Busy

Error 11 Missing Record

Error 12 Can't Find File

For information on customer support, please refer to the Introduction of this document.

Sample Queries and Databases

This chapter contains a sample database, followed by sample queries with the tables that are displayed by these queries.

For each query, wherever possible, both the EQL command and the Query Framer selections are shown. However, the Query Framer cannot produce some of the more sophisticated queries, which require the full capabilities of the EQL command language.

If you want to try these queries in Thoroughbred Query-IV, you may do so. The sample database is installed with Thoroughbred Query-IV and can be used to write your own queries or to test the queries presented in this chapter.

Generate sample queries

Before you can display the sample queries you must select **Generate Sample Files** from the System Administration Menu. For information see the Dictionary-IV Administrator Guide.

Sample Database

The queries in this chapter are based on the following two databases. The first database contains employee information using two tables (Q4EMPLOY and Q4DEPT). The second database contains customer information (UTCUST, UTREP, UTOPENAR, and UTSALHSH).

It is likely that access to data in the employee file would be restricted in a real-world situation. Thoroughbred IDOL-IV allows you to secure the data file (Link) with a password, to restrict specific terminals or operators, and to restrict display of individual Data Names (refer to security in the Dictionary-IV Developer Guide). For the purposes of illustrating the sample queries in this chapter, the sample data files are not secured.

Employee Database

Employee (Q4EMPLOY)

Employee Code	Employee Name	Job Description	Dept Code	Salary	Bonus
0001	Douglas	Bookkeeper	01	14500	
0002	Roland	Programmer	02	28750	1500
0003	Bell	Marketing Rep.	11	32000	2800
0004	Parker	Programmer	02	35000	1250
0005	Watson	Marketing Rep.	09	33000	1800
0006	Alexander	Sales Rep. (Regional)	05	30000	18000
0007	Drummond	Programmer	02	28500	1500
0008	Johnson	Sales Rep.	03	20000	8500
0009	Mason	Sales Rep. (Regional)	06	30000	17500
0010	Brady	Sales Rep.	03	20000	6750
0011	Perkins	Sales Rep.	03	20000	8800
0012	Allen	Sales Rep. (Regional)	07	30000	1950
0013	Davis	Bookkeeper	01	14500	

Department (Q4DEPT)

Dept. Code	Dept. Name	Dept. Location
01	Accounting	New York
02	Engineering	New York
03	Sales	New York
04	Marketing	New York
05	Sales	Atlanta
06	Sales	Chicago
07	Sales	San Francisco
08	Sales	London
09	Marketing	London
10	Sales	Toronto
11	Marketing	Toronto

Customer Database

Customer File (UTCUST)

Cust Code	Cust Name	Address	City	State	ZIP	Phone	Sales Rep.	Terms	Credit Limit	Credit Cmnt	Open AR Balance	YTD Sales
0001	Warren Baseball Club	1500 Boulevard Rd.	Warren	CT	10078	(203) 774-2415	003	6	1000	T		1618.55
0002	Lakewood Little League	40 North Main Street	South Hampton	NY	10112	(516) 934-0221	001	C	1000	T	822.50	2388.50
0003	Southside Raiders	100 Locust Ave.	Rochelle Park	NJ	07011	(201) 345-3240	004	3	1000	T	412.78	1153.65
0004	Blue Diamond Baseball	91 Bayway Ave.	Brooklyn	NY	10114	(718) 230-2424	002	6	1000	T	1105.52	1980.50
0005	Team Sports	503 Glenview Dr.	New York	NY	10012	(212) 423-4555	001	9	1000	T	664.05	2831.65
0006	Westfield Recreation	100 East Broad St.	Westfield	NJ	07090	(732) 232-1500	004	3	1000	T	525.97	2785.80
0007	Dayton Little Devils Baseball	725 Austin St.	Springfield	NJ	07082	(732) 433-4300	001	C	1000	T	503.70	1259.25
0008	Hartford Recreation Office	100 Route 66	Hartford	CT	10081	(203) 340-3434	003	6	1000	T	1088.25	2353.90
0009	Paramus Little League	750 Walsh Lane	Paramus	NJ	07231	(201) 236-4001	003	6	1000	T	291.30	1717.00
0010	Fords Little League All-Stars	1967 North Clarl St.	Fords	NJ	07034	(732) 345-9292	002	3	1000	T	347.55	505.85
0011	Princeton Amateur League	440 Nassau St.	Princeton	NJ	08718	(609) 359-9090	004	9	1000	T	687.50	2545.20
0012	Hackensack Hackers	191 Lincoln Drive	Hackensac	NJ	07071	(201) 452-3544	001	C	1000	T	825.21	1729.10

Rep Code (UTREP)

Rep Code	Rep Name	Phone Ext	YTD Commission
001	James Johnson	4513	546.05
002	Helen Tyler	4511	428.62
003	Mike Walsh	4518	235.60
004	Brian Andrew	4503	298.71

Open AR File (UTOPENAR)

Invoice Number	Due Date	Net Amount	Total Payments	Customer Code
B00102	02/05/98	664.05	398.43	0006
B00105	01/18/98	167.85	.00	0012
B00111	04/03/98	470.05	282.03	0004
B00115	02/14/98	165.85	.00	0002
B00117	02/19/98	448.20	268.92	0007
B00118	02/21/98	656.65	393.99	0012
B00123	04/29/98	493.50	.00	0004
B00128	03/08/98	811.05	486.63	0007
B00129	04/09/98	620.95	372.57	0003
B00135	06/19/98	300.20	.00	0011
B00136	03/24/98	314.55	.00	0002
B00137	03/24/98	394.70	.00	0012
B00138	05/28/98	199.70	.00	0004
B00139	05/29/98	668.25	.00	0008
B00140	06/04/98	291.30	.00	0009
B00141	05/06/98	347.55	.00	0010
B00142	07/06/98	387.75	.00	0011
B00143	06/10/98	224.30	.00	0004
B00144	07/12/98	664.05	.00	0005
B00145	05/15/98	260.35	.00	0006
B00146	06/17/98	420.00	.00	0008
B00147	04/18/98	342.10	.00	0002
B00148	05/19/98	164.40	.00	0003

Sales History (UTSALHSH)

Invoice Number	Invoice Date	Rep Code	Customer Code	Rep Commission	Net Sale Amount
B00100	01/05/98	004	0001	19.29	385.70
B00101	01/05/98	002	0005	56.02	700.25
B00102	01/06/98	001	0006	53.12	664.05
B00103	01/12/98	002	0011	56.74	709.25
B00104	01/15/98	003	0002	27.67	553.40
B00105	01/18/98	002	0012	13.43	167.85
B00106	01/19/98	001	0004	17.51	218.85
B00107	01/21/98	001	0008	09.03	112.90
B00108	01/24/98	002	0009	40.04	500.50
B00109	01/29/98	001	0010	12.66	158.30
B00110	02/01/98	004	0011	15.38	307.65

Invoice Number	Invoice Date	Rep Code	Customer Code	Rep Commission	Net Sale Amount
B00111	02/02/98	001	0004	37.60	470.05
B00112	02/05/98	002	0005	13.27	165.85
B00113	02/06/98	004	0006	32.91	658.10
B00114	02/08/98	004	0008	19.11	382.15
B00115	02/14/98	003	0002	08.29	165.85
B00116	02/15/98	004	0003	18.42	268.30
B00117	02/19/98	003	0007	22.41	448.20
B00118	02/21/98	001	0012	52.53	656.65
B00119	02/23/98	004	0002	50.63	1012.60
B00120	02/25/98	003	0011	42.02	840.35
B00121	02/27/98	004	0012	25.50	509.90
B00122	02/27/98	002	0005	22.79	284.85
B00123	02/28/98	004	0004	24.68	493.50
B00124	03/02/98	002	0004	29.93	374.10
B00125	03/04/98	002	0009	74.02	925.20
B00126	03/05/98	001	0008	61.65	770.60
B00127	03/05/98	002	0006	52.03	650.40
B00128	03/08/98	001	0007	64.88	811.05
B00129	03/10/98	003	0003	31.05	620.95
B00130	03/14/98	004	0001	26.15	523.05
B00131	03/14/98	001	0005	32.33	404.10
B00132	03/15/98	001	0001	56.78	709.80
B00133	03/16/98	004	0005	30.63	612.55
B00134	03/18/98	003	0006	27.65	552.90
B00135	03/21/98	004	0011	15.01	300.20
B00136	03/24/98	003	0002	15.73	314.55
B00137	03/24/98	002	0012	31.58	394.70
B00138	03/29/98	003	0004	09.99	199.70
B00139	03/30/98	003	0008	33.41	668.25
B00140	04/05/98	001	0009	23.30	291.30
B00141	04/06/98	003	0010	17.38	347.55
B00142	04/07/98	001	0011	31.02	387.75
B00143	04/11/98	002	0004	17.94	224.30
B00144	04/13/98	001	0005	53.12	664.05
B00145	04/15/98	002	0006	20.83	260.35
B00146	04/18/98	004	0008	21.00	420.00
B00147	04/18/98	001	0002	27.37	342.10
B00148	04/18/98	001	0003	13.15	164.40

Sample Queries

Q4S01

Selecting columns from a file and using a numeric output mask.

Request in English:

List all employees and their salaries.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, SALARY (Mask: $##,## 0)

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,## 0"
FROM Q4EMPLOY
```

Table Displayed:

Employee Name	Salary
Douglas	$14,500
Roland	$28,750
Bell	$32,000
Parker	$35,000
Watson	$33,000
Alexander	$30,000
Drummond	$28,500
Johnson	$28,500
Mason	$30,000
Brady	$20,000
Perkins	$20,000
Allen	$30,000
Davis	$14,500

Q4S02

Selecting columns from a file and specifying column headings.

Request in English:

List all employees and their salaries, using the specified column headings.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME (Heading: Name), SALARY (Heading: Salary; Mask: $##,## 0)

EQL Command:

```
SELECT EMPLOYEE-NAME "Name", SALARY "Salary"
:"$##,##0"
FROM Q4EMPLOY
```

Table Displayed:

Name	Salary
Douglas	$14,500
Roland	$28,750
Bell	$32,000
Parker	$35,000
Watson	$33,000
Alexander	$30,000
Drummond	$28,500
Johnson	$20,000
Mason	$30,000
Brady	$20,000
Perkins	$20,000
Allen	$30,000
Davis	$14,500

Q4S03

Selecting specific rows using WHERE, with rows matching (=) a value.

Request in English:

List the names and salaries of employees who earn $20,000.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where: SALARY = 20000

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE SALARY = 20000
```

Table Displayed:

Employee Name	Salary
Johnson	$20,000
Brady	$20,000
Perkins	$20,000

Q4S04

Selecting specific rows using WHERE, with rows greater than (>) a value.

Request in English:

List the names and salaries of employees who earn more than $20,000.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where: SALARY > 20000

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE SALARY > 20000
```

Table Displayed:

Employee Name	Salary
Roland	$28,750
Bell	$32,000
Parker	$35,000
Watson	$33,000
Alexander	$30,000
Drummond	$28,500
Mason	$30,000
Allen	$30,000

Q4S05

Selecting specific rows using WHERE, with rows matching a partial value (substring).

Request in English:

List the names and salaries of all programmers.

Query Framer:

Files (Links):	Q4EMPLOY
Columns:	EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where:	JOB-DESCRIPTION(1, 4) = "Prog"

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE JOB-DESCRIPTION(1,4) = "Prog"
```

Table Displayed:

Employee Name	Salary
Roland	$28,750
Parker	$35,000
Drummond	$28,500

Q4S06

Selecting specific rows using WHERE, with rows matching a partial value (LIKE).

Request in English:

List the names and salaries of all sales representatives.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where: JOB-DESCRIPTION LIKE "Sales Rep*"

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE JOB-DESCRIPTION LIKE "Sales Rep*"
```

Table Displayed:

Employee Name	Salary
Alexander	$30,000
Johnson	$20,000
Mason	$30,000
Brady	$20,000
Perkins	$20,000
Allen	$30,000

Q4S07

Selecting specific rows using WHERE, with rows matching a partial value (LIKE).

Request in English:

List the names and salaries of only regional sales representatives.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where: JOB-DESCRIPTION LIKE "*Regional*"

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE JOB-DESCRIPTION LIKE "*Regional*"
```

Table Displayed:

Employee Name	Salary
Alexander	$30,000
Mason	$30,000
Allen	$30,000

Q4S08

Selecting specific rows using WHERE, with rows matching more than 1 value (AND).

Request in English:

List the names and salaries of programmers who earn more than $30,000.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where: JOB-DESCRIPTION LIKE "Prog*" AND SALARY> 30000

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE JOB-DESCRIPTION LIKE "Prog*" AND SALARY > 30000
```

Table Displayed:

Employee Name	Salary
Parker	$35,000

Q4S09

Selecting specific rows using WHERE, with rows meeting either of 2 values (OR).

Request in English:

List the names and salaries of all employees who earn more than $30,000 and of all programmers.

Query Framer:

Files (Links):	Q4EMPLOY
Columns:	EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where:	JOB-DESCRIPTION(1,4) = "Prog" OR SALARY > 30000

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE JOB-DESCRIPTION(1,4) = "Prog" OR SALARY > 30000
```

Table Displayed:

Employee Name	Salary
Roland	$28,750
Bell	$32,000
Parker	$35,000
Watson	$33,000
Drummond	$28,500

Q4S10

Selecting specific rows using WHERE, when rows are BETWEEN 2 values.

Request in English:

List the names and salaries of employees who earn between $20,000 and $30,000.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where: SALARY BETWEEN 20000 AND 30000

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE SALARY BETWEEN 20000 AND 30000
```

Table Displayed:

Employee Name	Salary
Roland	$28,750
Alexander	$30,000
Drummond	$28,500
Johnson	$20,000
Mason	$30,000
Brady	$20,000
Perkins	$20,000
Allen	$30,000

Q4S11

Selecting specific rows using WHERE, with rows matching a partial value (LIKE).

Request in English:

List the names and salaries of employees whose names begin with a □P□ and have □rk□ as the second and third letters.

Query Framer:

Files (Links):	Q4EMPLOY
Columns:	EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where:	EMPLOYEE-NAME LIKE "P?rk*"

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE EMPLOYEE-NAME LIKE "P?rk*"
```

Table Displayed:

Employee Name	Salary
Parker	$35,000
Perkins	$20,000

Q4S12

Selecting specific rows using WHERE, with rows matching a partial value (LIKE).

Request in English:

List the names and salaries of employees whose names begin with a □P□, □M□, or □W□.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Rows Where: EMPLOYEE-NAME LIKE "[PMW]*"

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0"
FROM Q4EMPLOY
WHERE EMPLOYEE-NAME LIKE "[PMW]*"
```

Table Displayed:

Employee Name	Salary
Parker	$35,000
Watson	$33,000
Mason	$30,000
Perkins	$20,000

Q4S13

Selecting specific rows using WHERE, with rows matching values IN a list.

Request in English:

List the names, salaries, and bonus amounts of employees who received a bonus of $1,500 or $1,250.

Query Framer:

Files (Links):	Q4EMPLOY
Columns:	EMPLOYEE-NAME, SALARY
	(Mask: $##,## 0),BONUS
Rows Where:	BONUS IN (1500, 1250)

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0",BONUS
FROM Q4EMPLOY
WHERE BONUS IN (1500, 1250)
```

Table Displayed:

Employee Name	Salary	Bonus
Roland	$28,750	1500
Parker	$35,000	1250
Drummond	$28,500	1500

Q4S14

Selecting specific rows using WHERE, with rows matching values NOT IN a list.

Request in English:

List the names, salaries, and bonus amounts of employees who did not receive a bonus of $1,500 or $1,250.

Query Framer:

Files (Links):	Q4EMPLOY
Columns:	EMPLOYEE-NAME, SALARY
	(Mask: $##,## 0),BONUS
Rows Where:	BONUS NOT IN (1500, 1250)

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY :"$##,##0", BONUS
FROM Q4EMPLOY
WHERE BONUS NOT IN (1500, 1250)
```

Table Displayed:

Employee Name	Salary	Bonus
Douglas	$14,500	
Bell	$32,000	2800
Watson	$33,000	1800
Alexander	$30,000	18000
Johnson	$20,000	8500
Mason	$30,000	17500
Brady	$20,000	6750
Perkins	$20,000	8800
Allen	$30,000	19500
Davis	$14,500	

Q4S15

Selecting a calculated column.

Request in English:

List the names and gross pay (including bonus) of marketing representatives.

Query Framer:

Use EQL for calculated columns.

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY + BONUS :"$##,##0"
FROM Q4EMPLOY
WHERE JOB-DESCRIPTION LIKE "Marketing*"
```

Table Displayed:

Employee Name	Salary + Bonus
Bell	$34,800
Watson	$34,800

Q4S16

Selecting a calculated column using an expression alias.

Request in English:

List the names and gross pay (including bonus) of non-regional sales representatives.

Query Framer:

Use EQL for calculated columns.

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY+BONUS
        AS GROSS-PAY:"$##,##0"
FROM Q4EMPLOY
WHERE JOB-DESCRIPTION LIKE "Sales*"
        AND JOB-DESCRIPTION NOT LIKE "*Regional*"
```

Table Displayed:

Employee Name	Gross Pay
Johnson	$28,500
Brady	$26,750
Perkins	$28,800

Q4S17

Selecting a calculated column and specific rows using an expression alias.

Request in English:

List the names and gross pay (including bonus) of employees who earned $35,000 or more.

Query Framer:

Use EQL for calculated columns.

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY + BONUS
        AS GROSS-PAY :"$##,##0"
FROM Q4EMPLOY
WHERE GROSS-PAY >= 35000
```

Table Displayed:

Employee Name	Gross Pay
Parker	$36,250
Alexander	$48,000
Mason	$47,500
Allen	$49,500

Q4S18

Producing a TOTAL of numeric columns using an expression alias.

Request in English:

Show the effect of an □across-the-board□ salary increase of 7% for employees, listing the current salary, amount of salary increase, the new salary, and the grand total for each of these figures.

Query Framer:

Use EQL for calculated columns.

EQL Command:

```
TOTAL SALARY, PAY-RAISE, NEW-PAY;
SELECT EMPLOYEE-NAME, SALARY :"###,###"", SALARY *.07
        AS PAY-RAISE :"###,###",SALARY * 1.07 AS
        NEW-PAY :"###,###"
FROM Q4EMPLOY
```

Table Displayed:

Employee Name	Salary	Pay Raise	New Pay
Douglas	14,500	1,015	15,515
Roland	28,750	2,013	30,763
Bell	32,000	2,240	34.240
Parker	35,000	2,450	37,450
Watson	33,000	2.310	35,310
Alexander	30,000	2,100	32,100
Drummond	28,500	1,994	30,495
Johnson	20,000	1,400	21,400
Mason	30,000	2,100	32,100
Brady	20,000	1,400	21,400
Perkins	20,000	1,400	21,400
Allen	30,000	2,100	32,100
Davis	14,500	1,015	15,515
	336,250	23,538	359,788

Q4S19

Displaying rows in ORDER BY a specific column.

Request in English:

List the names and salaries of all employees in alphabetical order.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, SALARY (Mask: $##,## 0)
Order Rows By: SORT1 EMPLOYEE-NAME

EQL Command:

```
SELECT EMPLOYEE-NAME, SALARY: "$##,##0"
FROM Q4EMPLOY
ORDER BY EMPLOYEE-NAME
```

Table Displayed:

Employee Name	Salary
Alexander	$30,000
Allen	$30,000
Bell	$32,000
Brady	$20,000
Davis	$14,500
Douglas	$14,500
Drummond	$28,500
Johnson	$20,000
Mason	$30,000
Parker	$35,000
Perkins	$20,000
Roland	$28,750
Watson	$33,000

Q4S20

Displaying rows in descending, or reverse, ORDER BY a specific column.

Request in English:

List the names and bonus amounts of all employees, going from the employee who received the biggest bonus amount to the one who received the smallest.

Query Framer:

Files (Links): Q4EMPLOY
Columns: EMPLOYEE-NAME, BONUS (Mask: $##,## 0)
Order Rows By: SORT2 BONUS(D)

EQL Command:

```
SELECT EMPLOYEE-NAME, BONUS :"$##,##0"
FROM Q4EMPLOY
ORDER BY BONUS(D)
```

Table Displayed:

Employee Name	Bonus
Allen	$19500
Alexander	$18000
Mason	$17500
Perkins	$8800
Johnson	$8500
Brady	$6750
Bell	$2800
Watson	$1800
Roland	$1500
Drummond	$1500
Parker	$1250
Douglas	$0
Davis	$0

Q4S21

Displaying rows in ORDER BY 2 columns.

Request in English:

Show all job titles in alphabetical order, giving the range of salaries for each position going from the lowest salary to the highest.

Query Framer:

Files (Links):	Q4EMPLOY
Columns:	JOB-DESCRIPTION, SALARY
	(Mask: $##,## 0), EMPLOYEE-NAME
Order Rows By:	SORT3 JOB-DESCRIPTION + SALARY

EQL Command:

```
SELECT JOB-DESCRIPTION, SALARY :"$##,##0", EMPLOYEE-
NAME
FROM Q4EMPLOY
ORDER BY JOB-DESCRIPTION + SALARY
```

Table Displayed:

Job Description	Salary	Employee Name
Bookkeeper	$14,500	Douglas
Bookkeeper	$14,500	Davis
Marketing Rep.	$32,000	Bell
Marketing Rep.	$33,000	Watson
Programmer	$28,500	Drummond
Programmer	$28,750	Roland
Programmer	$35,000	Parker
Sales Rep.	$20,000	Johnson
Sales Rep.	$20,000	Brady
Sales Rep.	$20,000	Perkins
Sales Rep. (Regional)	$30,000	Alexander
Sales Rep. (Regional)	$30,000	Mason
Sales Rep. (Regional)	$30,000	Allen

Q4S22

Producing a TOTAL of a numeric column.

Request in English:

List all salaries and give the grand total.

Query Framer:

Files (Links): Q4EMPLOY
Columns: SALARY (Mask: $###,## 0)
Column Totals: SALARY

EQL Command:

```
TOTAL SALARY;
SELECT SALARY :"$###,##0"
FROM Q4EMPLOY
```

Table Displayed:

Salary
$ 14,500
$ 28,750
$ 32,000
$ 35,000
$ 33,000
$ 30,000
$ 28,500
$ 20,000
$ 30,000
$ 20,000
$ 20,000
$ 30,000
$ 14,500
$336,250

Q4S23

Producing the SUM of a numeric column.

Request in English:

Only show the total of all salaries.

Query Framer:

Use EQL to produce the SUM for all rows.

EQL Command:

```
SELECT SUM(SALARY) :"$###,##0"
FROM Q4EMPLOY
```

Table Displayed:

Sum (Salary)
$336,250

Q4S24

Producing the SUM of a numeric column while selecting only specific rows to be included in the SUM.

Request in English:

Show the total of salaries paid to the bookkeepers.

Query Framer:

Use EQL to produce the SUM for all selected rows.

EQL Command:

```
SELECT SUM(SALARY) :"$###,##0"
FROM Q4EMPLOY
WHERE JOB-DESCRIPTION LIKE "Bookkeeper*"
```

Table Displayed:

Sum (Salary)
$29,000

Q4S25

Summarizing a group of records: selecting a group (GROUP BY) and producing a SUM for each group.

Request in English:

List the total of salaries in each department.

Query Framer:

Files (Links): Q4EMPLOY
Columns: DEPT-CODE, SALARY (Mask: $###,## 0)
Group Rows By: DEPT-CODE

EQL Command:

```
SELECT DEPT-CODE, SUM(SALARY) :"$###,##0"
FROM Q4EMPLOY
GROUP BY DEPT-CODE
```

Table Displayed:

Dept. Code	Sum (salary)
01	$29,000
02	$92,250
03	$60,000
05	$30,000
06	$30,000
07	$30,000
09	$33,000
11	$32,000

Q4S26

Summarizing a group of records: selecting a group (GROUP BY) and producing a SUM for each group (with 2 files).

Request in English:

List the total salary for each department, showing departments by name.

Query Framer:

Files (Links): Q4DEPT, Q4EMPLOY
Columns: DEPT-NAME, SALARY (Mask: $###,## 0)
Group Rows By: DEPT-NAME

EQL Command:

```
SELECT DEPT-NAME, SUM(SALARY) :"$###,##0"
FROM Q4DEPT, Q4EMPLOY
GROUP BY DEPT-NAME
```

Table Displayed:

Dept. Name	Sum (Salary)
Accounting	$ 29,000
Engineering	$ 92,250
Marketing	$ 65,000
Sales	$150,000

Q4S27

Selecting columns from 2 related files.

Request in English:

List departments alphabetically, showing the employees within each department.

Query Framer:

Files (Links): Q4DEPT, Q4EMPLOY
Columns: DEPT-NAME, EMPLOYEE-NAME
Order Rows By: SORT1 DEPT-NAME

EQL Command:

```
SELECT DEPT-NAME, EMPLOYEE-NAME
FROM Q4DEPT, Q4EMPLOY
ORDER BY DEPT-NAME
```

Table Displayed:

Dept. Name	Employee Name
Accounting	Douglas
Accounting	Davis
Engineering	Roland
Engineering	Parker
Engineering	Drummond
Marketing	Watson
Marketing	Bell
Sales	Johnson
Sales	Brady
Sales	Perkins
Sales	Alexander
Sales	Mason
Sales	Allen

Q4S28

Selecting columns from 2 related files.

Request in English:

List department locations alphabetically, showing the employees and their jobs within each department.

Query Framer:

Files (Links):	Q4DEPT, Q4EMPLOY
Columns:	DEPT-LOCATION, EMPLOYEE-NAME, JOB-DESCRIPTION
Order Rows By:	SORT2 DEPT-LOCATION

EQL Command:

```
SELECT DEPT-LOCATION, EMPLOYEE-NAME, JOB-DESCRIPTION
FROM Q4DEPT, Q4EMPLOY
ORDER BY DEPT-LOCATION
```

Table Displayed:

Dept. Location	**Employee Name**	**Job Description**
Atlanta	Alexander	Sales Rep (Regional)
Chicago	Mason	Sales Rep (Regional)
London	Watson	Marketing Rep
New York	Douglas	Bookkeeper
New York	Davis	Bookkeeper
New York	Roland	Programmer
New York	Parker	Programmer
New York	Drummond	Programmer
New York	Johnson	Sales Rep
New York	Brady	Sales Rep
New York	Perkins	Sales Rep
San Francisco	Allen	Sales Rep (Regional)
Toronto	Bell	Marketing Rep

Q4S29

Producing the SUM of a calculated column from one file while selecting specific rows matching the value from a second file.

Request in English:

List the total of gross pay (including bonus) for employees who work in New York.

Query Framer:

Use EQL to produce the SUM of a calculated column.

EQL Command:

```
SELECT SUM(SALARY + BONUS) :"$###,##0"
FROM Q4DEPT, Q4EMPLOY
WHERE DEPT-LOCATION LIKE "New York*"
```

Table Displayed:

Sum (Salary + Bonus)
$209,550

Q4S30

Producing a list ordered by one Data Name then ordered by a second Data Name.

Request in English:

Provide a telephone list of customers within each state and sorted by customer name.

Query Framer:

Files (Links):	UTCUST
Columns:	CUST-NAME, STATE, PHONE (Mask: ### ###-####)
Order Rows By:	STATE, CUST-NAME

EQL Command:

```
SELECT CUST-NAME, STATE, PHONE :"### ###-####"
FROM UTCUST
ORDER ROWS BY STATE, CUST-NAME
```

Table Displayed:

Customer Name	State	Telephone
Hartford Recreation Office	Connecticut	203 340-3434
Warren Baseball Club	Connecticut	203 774-2415
Dayton Little Devils Baseball	New Jersey	732 433-4300
Fords Little League All-Stars	New Jersey	732 345-9292
Hackensack Hackers	New Jersey	201 452-3544
Paramus Little League	New Jersey	201 236-4001
Princeton Amateur League	New Jersey	609 359-9090
Southside Raiders	New Jersey	201 345-3240
Westfield Recreation	New Jersey	732 232-1500
Blue Diamond Baseball	New York	718 230-2424
Lakewood Little League	New York	516 934-0221
Team Sports	New York	212 423-4555

Q4S31

Producing a list from two files which is ordered by one Data Name then ordered by a second Data Name.

Request in English:

Provide a list of invoices for each customer listing the most current invoice first.

Query Framer:

Files (Links):	UTSALHSH, UTCUST
Columns:	CUST-NAME, INVOICE-NUMBER, INVOICE-DATE, NET-SALE-AMOUNT (Mask: $#######.00)
Order Rows By:	CUST-NAME, INVOICE-DATE (Descending order)

EQL Command:

```
SELECT CUST-NAME, INVOICE-NUMBER, INVOICE-DATE,
        NET-SALE-AMOUNT :"$#######.00"
FROM UTSALHSH, UTCUST
ORDER ROWS BY CUST-NAME, INVOICE-DATE(D)
```

Table Displayed:

Customer Name	Invoice Number	Invoice date	Net Sales Amount
Warren Baseball Club	B00132	05/15/98	709.80
	B00130	03/14/98	523.05
	B00100	01/05/98	385.70
Lakewood Little League	B00147	04/18/98	342.10
	B00136	03/24/98	314.55
	B00119	02/23/98	1012.60
	B00115	02/14/98	165.85
	B00104	01/15/98	553.40
Southside Raiders	B00148	04/19/98	164.40
	B00129	03/10/98	620.95
	B00116	02/15/98	368.30
Blue Diamond Baseball	B00143	01/11/98	224.30
	B00138	03/29/98	199.70
	B00124	03/02/98	374.10
	B00123	02/28/98	493.50
	B00111	02/02/98	470.05
	B00106	01/19/98	218.85
Team Sports	B00144	04/13/98	664.05
	B00133	03/16/98	612.55
	B00131	03/14/98	404.10
	B00122	02/27/98	284.85
	B00112	02/05/98	165.85
	B00101	01/05/98	700.25
Westfield Recreation	B00145	04/15/98	260.35
	B00134	03/18/98	552.90
	B00127	03/05/98	650.40
	B00113	02/06/98	658.10
	B00102	01/06/98	664.05
Dayton Little Devils Baseball	B00128	03/08/98	811.05
	B00117	02/19/98	448.20

Customer Name	Invoice Number	Invoice date	Net Sales Amount
Hartford Recreation Office	B00146	04/18/98	420.00
	B00139	03/30/98	668.25
	B00126	03/05/98	770.60
	B00114	02/08/98	382.15
	B00107	01/21/98	112.90
Paramus Little League	B00140	04/05/98	291.30
	B00125	03/04/98	925.20
	B00108	01/24/98	500.50
Fords Little League All-Stars	B00141	04/05/98	347.55
	B00109	01/29/98	158.30
Princeton Amateur League	B00142	04/07/98	387.75
	B00135	03/21/98	300.20
	B00120	02/25/98	840.35
	B00110	02/01/98	307.65
	B00103	01/12/98	709.25
Hackensack Hackers	B00137	03/24/98	394.70
	B00121	02/27/98	509.90
	B00118	02/21/98	656.65
	B00105	01/18/98	167.85

Q4S32

Producing a list using a mask for specified columns.

Request in English:

Provide a list of open receivables and payments toward each invoice.

Query Framer:

Files (Links): UTOPENAR
Columns: INVOICE-NUMBER, DUE-DATE, CUST-CODE, NET-AMOUNT (Mask: $#######.00), TOTAL-PAYMENTS (Mask: $#######.00)

EQL Command:

```
SELECT INVOICE-NUMBER,DUE-DATE,CUST-CODE,NET-AMOUNT
      :"$#######.00",TOTAL-PAYMENTS:"$#######.00"
FROM UTOPENAR
```

Table Displayed:

Invoice Number	Due Date	Customer Code	Net Amount	Total Payments
B00102	02/05/98	0006	$664.05	$398.43
B00105	01/18/98	0012	$167.85	$ .00
B00111	04/03/98	0004	$470.05	$282.03
B00115	02/14/98	0002	$165.85	$ .00
B00117	02/19/98	0007	$448.20	$268.92
B00118	02/21/98	0012	$656.65	$393.99
B00123	04/29/98	0004	$493.50	$ .00
B00128	03/08/98	0007	$811.05	$486.63
B00129	04/09/98	0003	$620.95	$372.57
B00135	06/19/98	0011	$300.20	$ .00
B00136	03/24/98	0002	$314.55	$ .00
B00137	03/24/98	0012	$394.70	$ .00
B00138	05/28/98	0004	$199.70	$ .00
B00139	05/29/98	0008	$668.25	$ .00
B00140	06/04/98	0009	$291.30	$ .00
B00141	05/06/98	0010	$347.55	$ .00
B00142	07/06/98	0011	$387.75	$ .00
B00143	06/10/98	0004	$224.30	$ .00
B00144	07/12/98	0005	$664.05	$ .00
B00145	05/15/98	0006	$260.35	$ .00
B00146	06/17/98	0008	$420.00	$ .00
B00147	04/18/98	0002	$342.10	$ .00
B00148	05/19/98	0003	$164.40	$ .00

Q4S33

Request in English:

Provide a list of invoices and the current balance by customer.

Query Framer:

Use EQL to produce the SUM of a calculated column.

EQL Command:

```
DETAIL
TOTAL SUM(NET-AMOUNT-TOTAL-PAYMENTS);
SELECT CUST-NAME,INVOICE-NUMBER,SUM(NET-AMOUNT-
        TOTAL-PAYMENTS)"Invoice|Balance"
FROM UTOPENAR,UTCUST
GROUP BY CUST-NAME
```

Table Displayed:

Customer Name	Invoice Number	Invoice Balance
Lakewood Little League	3000	700.00
		700.00
Warren Baseball Club	1000	70.00
Warren Baseball Club	2000	300.00
		370.00

Q4S34

Producing a list that allows the user to determine the specific group to display.

Request in English:

Provide a customer list with telephone for any sales-rep. You determine the sales-rep at run time.

Query Framer:

Use EQL to produce the SUM of a calculated column.

EQL Command:

```
INPUT "Enter Sales Rep:"INTO TEMP-REP(3);
SELECT CUST-CODE,CUST-NAME,PHONE
FROM UTCUST,UTREP
WHERE REP-CODE=TEMP-REP
```

User Prompt:

Enter Sales Rep: 003

Table Displayed:

Customer Code	Customer Name	Telephone
0001	Warren Baseball Club	203 774-2415
0008	Hartford Recreation Office	203 340-3434
0009	Paramus Little League	201 236-4001

Q4S35

Producing a list which resizes a column.

Request in English:

Provide a customer list of open AR balances. Only display the first 20 characters of the customer name.

Query Framer:

Files (Links): UTCUST
Columns: CUST-CODE, CUST-NAME (only first 20 characters), OPEN-AR-BALANCE (Mask: $######.00)

EQL Command:

```
SELECT CUST-CODE,CUST-NAME(1,20),
        OPEN-AR-BALANCE:"$######.00"
FROM UTCUST
```

Table Displayed:

Customer Code	Customer Name	Open AR Balance
0001	Warren Baseball Club	
0002	Lakewood Little League	$822.50
0003	Southside Raiders	$412.78
0004	Blue Diamond Baseball	$1105.52
0005	Team Sports	$664.05
0006	Westfield Recreation	$525.97
0007	Dayton Little Devils	$503.70
0008	Hartford Recreation	$1088.25
0009	Paramus Little League	$291.30
0010	Fords Little League	$347.55
0011	Princeton Amateur League	$687.95
0012	Hackensack Hackers	$825.21

Q4S36

Producing a list which fit a specified criteria.

Request in English:

Provide a list of invoices for March of 1998.

Query Framer:

Files (Links):	UTSALHSH
Columns:	INVOICE-NUMBER, CUST-CODE, INVOICE-DATE, NET-SALE-AMOUNT (Mask: $#######.00)
Rows Where:	INVOICE-DATE(YYMM) = "9803"

EQL Command:

```
SELECT INVOICE-NUMBER,CUST-CODE,INVOICE-DATE
     NET-SALE-AMOUNT:"$#######.00"
FROM UTSALHSH
ROWS WHERE INVOICE-DATE(YYMM)="9803"
```

Table Displayed:

Invoice Number	Customer Code	Invoice Date	Net Sales Amount
B00124	0004	03/02/98	$374.10
B00125	0009	03/04/98	$925.20
B00126	0008	03/05/98	$770.60
B00127	0006	03/05/98	$650.40
B00128	0007	03/08/98	$4811.05
B00129	0003	03/10/98	$620.95
B00130	0001	03/14/98	$523.05
B00131	0005	03/14/98	$404.10
B00132	0001	03/15/98	$709.80
B00133	0005	03/16/98	$612.55
B00134	0006	03/18/98	$552.90
B00135	0011	03/21/98	$300.20
B00136	0002	03/24/98	$314.55
B00137	0012	03/24/98	$394.70
B00138	0004	03/29/98	$199.70
B00139	0008	03/30/98	$668.25