Vault
Interactive Query Facility Guide

Optegra® Release 6

DOC35031-011

Parametric Technology Corporation

Information described in this document is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Registered Trademarks of Parametric Technology Corporation or a Subsidiary

Trademarks of Parametric Technology Corporation or a Subsidiary

Third-Party Trademarks

Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494-2714
8 January 2001

Table of Contents

Preface

Introduction

Getting Started

Creating and Using Queries in IQF

Formatting a Query Result

Filing and Printing a Query Result

Defining New Views for IQF Use

Glossary

Preface

The *Vault Interactive Query Facility Guide* explains how to view the contents of the Vault control tables for Optegra products. You can format and print reports based on the control table information. The *Vault Interactive Query Facility Guide* is for experienced Vault users who need access to information in the control tables.

Vault is a part of the Optegra object-oriented solutions for Enterprise Data Management. Vault uses a client/server architecture to provide cost-effective data management and release management capabilities for workstation and PC use.

Related Documents

The following documents may be helpful as you use *Vault Interactive Query Facilty Guide*:

- *Vault Database Tables*
- *Vault Command Reference*

Book Conventions

The following table illustrates and explains conventions used in writing about Optegra applications.

Convention	Example	Explanation
EPD_HOME	`cd $EPD_HOME/install` (UNIX) `cd %EPD_HOME%\install` (Windows)	Represents the default path where the current version of the product is installed.
Menu selections	Vault > Check Out > Lock	Indicates a command that you can choose from a menu.
Command buttons and options	Mandatory check box, Add button, Description text box	Names selectable items from dialog boxes: options, buttons, toggles, text boxes, and switches.

User input and code	`Wheel_Assy_details`
	`-xvf /dev/rst0`
	`Enter command>` **`plot_config`**
		Enter the text in a text box or on a command line.
		Where system output and user input are mixed, user input is in bold.

System output	`CT_struct.aename`	Indicates system responses.
Parameter and variable names	`tar -cvf /dev/rst0` filename	Supply an appropriate substitute for each parameter or variable; for example, replace filename with an actual file name.
Commands and keywords	The `ciaddobj` command creates an instance of a binder.	Shows command syntax.
Text string	`"SRFGROUPA"` or `'SRFGROUPA'`	Shows text strings. Enclose text strings with single or double quotation marks.
Integer	n	Supply an integer for *n.*
Real number	x	Supply a real number for *x.*
#	`# mkdir /cdrom`	Indicates the root (superuser) prompt on command lines.
%	`% rlogin` remote_system_name `-l root`	Indicates the C shell prompt on command lines.
$	`$ rlogin` remote_system_name `-l root`	Indicates the Bourne shell prompt on command lines.
>	`> copy` filename	Indicates the MS-DOS prompt on command lines.
Keystrokes	`Return` or `Control-g`	Indicates the keys to press on a keyboard.

Online User Documentation

Online documentation for each Optegra book is provided in HTML if the documentation CD-ROM is installed. You can view the online documentation from an HTML browser or from the HELP command.

You can also view the online documentation directly from the CD-ROM without installing it.

From an HTML Browser:

1. Navigate to the directory where the documents are installed. For example,

   $EPD_HOME`/data/html/htmldoc/` (UNIX)

   %EPD_HOME%`\data\html\htmldoc\` (Windows NT)
2. Click `mainmenu.html`. A list of available Optegra documentation appears.
3. Click the book title you want to view.

From the HELP Command:

To view the online documentation for your specific application, click HELP. (Consult the documentation specific to your application for more information.)

From the Documentation CD-ROM:

1. Mount the documentation CD-ROM.
2. Point your browser to:

   CDROM_mount_point`/htmldoc/mainmenu.html` (UNIX)

   CDROM_Drive:`\htmldoc\mainmenu.html` (Windows NT)

Printing Documentation

A PDF (Portable Document Format) file is included on the CD-ROM for each online book. See the first page of each online book for the document number referenced in the PDF file name. Check with your system administrator if you need more information.

You must have Acrobat Reader installed to view and print PDF files.

The default documentation directories are:

- $EPD_HOME`/data/html/pdf/`doc_number`.pdf` (UNIX)
- %EPD_HOME%`\data\html\pdf\`doc_number`.pdf` (Windows NT)

Resources and Services

For resources and services to help you with PTC (Parametric Technology Corporation) software products, see the *PTC Customer Service Guide*. It includes instructions for using the World Wide Web or fax transmissions for customer support.

Documentation Comments

PTC welcomes your suggestions and comments. You can send feedback in the following ways:

- Send comments electronically to `doc-webhelp@ptc.com.`
- Fill out and mail the PTC Documentation Survey located in the *PTC Customer Service Guide.*

Chapter 1

Introduction

This chapter introduces the concept of the Interactive Query Facility.

- Introduction to Interactive Query Facility

Introduction to Interactive Query Facility

The Interactive Query Facility (IQF) helps you view and generate reports of information stored in Vault.

Information in the database tables that Vault uses to manage data are called control tables. IQF uses these tables to generate results to queries for information. Vault control tables are standard Structured Query Language (SQL) tables. SQL is a standard language for retrieving information in a relational database management system (RDBMS). It is approved by the American National Standards Institute (ANSI).

Oracle is an example of an RDBMS that complies with the SQL standard. SQL*Plus is the interactive facility for Oracle (Oracle Corporation). IQF is the interactive facility for Vault.

Consequently, with IQF you can view information about users, different versions of files, projects, and authority schemes stored in Vault.

Use IQF to select the exact information you require and perform the following tasks:

- Save the information in a file.
- Print one or more copies of the information.
- Store the selection (query) in a local file so that you can request the same information again.

IQF includes a set of command files containing predefined queries. Each file contains commands that generate a typical report. Use these files as they are or modify them to meet your needs. You can also define and store queries to produce customized reports.

IQF does not have its own programmatic interface. Consult the documentation specific to your relational database management system for information about the programmatic interface it supports.

Chapter 2

Getting Started

This chapter helps you get started using the Interactive Query Facility.

- Entering and Exiting IQF
- Using IQF Commands
- Using Views in IQF
- Understanding IQF Messages
- Using Online Help
- Using Cases in IQF

Entering and Exiting IQF

To use IQF you must be a valid Vault user and have the IQF command in your command list.

To access IQF:

1. Sign on to Vault using the `cisignon` command at the operating system prompt.
2. Type `ciiqf` at the operating system prompt. The IQF prompt appears, as follows:

```
% cisignon
% ciiqf
IQF>
```

To exit IQF and return to the operating system prompt, type `EXIT` or `QUIT` at the IQF prompt as follows:

```
IQF> EXIT
%
```

Using IQF Commands

Use IQF to view specific information stored in the Vault control tables. A sample session illustrating an IQF session follows:

```
% ciiqf
IQF> SELECT USER_ID, PROJECT_ID, READ_AUTHORITY, -
IQF> WRITE_AUTHORITY FROM USER_AUTHORITY
IQF> FORMAT TTITLE 'USER AUTHORITIES'
IQF> FORMAT BTITLE 'COMPANY CONFIDENTIAL'
IQF> FILE USER.REPORT
CQFFIL126I Results have been filed in USER.REPORT.
IQF> TYPE USER.REPORT

31-Jan-10                                 USER AUTHORITIES
                                          PAGE   1

USER_ID    PROJECT_ID    READ_AUTHORITY    WRITE_AUTHORITY
-------    ----------    --------------    ---------------

EDMADMIN                            99                  99
ORVE                                99                  99
TRRVE                               99                  99
ORVE        RVE                     99                  99
EDMADMIN    RVE                     99                  99
EDMADMIN    NONEREV                 99                  99
                                      COMPANY CONFIDENTIAL
IQF> EXIT
%
```

The following are the salient features of the preceding sample session:

- The SELECT command specifies the column names of the information needed. The USER_ID, PROJECT_ID, READ_AUTHORITY, and WRITE_AUTHORITY column names are specified in the query. For details, refer to “Defining a Query Using the SELECT Command” on page 3-2.
- The FROM option is used to specify the view IQF must access to get the information stored in the Vault control tables. For details, refer to “Using Views in IQF” on page 2-7.
- The FORMAT command is used to format the result of a query. For details, refer to Chapter 4, “Formatting a Query Result.”
- To save the result of the query, execute the FILE command. This saves the result of a query in a file. For more information on displaying query results, refer to Chapter 5, “Filing and Printing a Query Result.”
- To view the results of a query, execute the TYPE command. This displays of the contents of the file. For more information, refer to “Displaying the Contents of a Predefined Query” on page 3-14.

Brief Definitions of IQF Commands

The following table provides a brief definition of each IQF command. `SELECT` is the only SQL command available in IQF. The other IQF commands provide a subset of the functions provided by the standard SQL commands.

Table 2-1 Brief Descriptions of IQF Commands

Command	Function
`BREAK`	Inserts one or more empty lines in the display of the query result. See also “Adding Breaks in a Query Result” on page 4-27.
`EDIT`	Changes the prompt to `EDT>` where you can type in multiple IQF commands and then execute them at the same time or store them in a file. See also “Using IQF Predefined Queries” on page 3-12.
`END`	Ends the display of a query result or help text. See also “Entering and Exiting IQF” on page 2-2. This command also exits the EDIT mode. See also “Ending an EDIT Session” on page 3-10.
`EXIT`	Terminates IQF and returns you to the operating system prompt. See also “Entering and Exiting IQF” on page 2-2.
`FILE`	Saves the query result in a local file. See also “Filing the Results of a Query” on page 5-2.
`FORMAT`	Allows you to format a query result. See also “Defining the Format of a Query Result” on page 4-2.
`HELP`	Displays online help for IQF. See also “Using Online Help” on page 2-12.
`LIST`	Displays the current query. See also “Displaying the Current Query” on page 3-7.
`PRINT`	Sends the result of the current query or the result of a previous query saved in a file to the local printer. See also “Printing the Results of a Query” on page 5-5.
`QUIT`	Performs the same function as the `EXIT` command. See also “Using IQF Commands” on page 2-3.
`RUN`	Executes the current query when it does not include a placeholder. See also “Executing a Query without Placeholders” on page 3-5.
`SAVE`	Saves the current query in a local file. See also “Saving the Current Query” on page 3-8. This command also saves a group of IQF commands created in the EDIT mode in a local file. See also “Using IQF Predefined Queries” on page 3-12.
`SELECT`	Specifies the exact information required and the tables from which to extract the information. See also “Defining a Query Using the SELECT Command” on page 3-2.
`SET`	Assigns values to certain format parameters. See also “Displaying the Value of Format Parameters” on page 4-25.
`SHOW`	Displays the current format parameter settings. See also “Setting Format Parameters for a Query Result” on page 4-18.
`START`	Executes the current query or a stored query. See also “Executing the Current Query” on page 3-5.
`TYPE`	Displays the contents of a local file. See also “Displaying the Contents of a Predefined Query” on page 3-14.

Conventions in IQF Commands

The following are the conventions for using IQF commands:

- The length of an IQF command line can be as long as your system input line.
- Commands continue for more than a single line. To continue a command on the next line, end the current line with a hyphen (**-**) and press `Enter`. IQF replaces the hyphen with the first character on the next line. For example, if you type:

  ```
  IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, -
  IQF> UPDATE_DATE FROM USERS
  ```

  IQF recognizes `UPDATE_DATE` as another column name. You can also break up the command as:

  ```
  IQF> SELECT USER_ID, LAST_NAME, USER_GRO-
  IQF> UP, UPDATE_DATE FROM USERS
  ```

- Put at least one space between keywords. The placement of the hyphen in the following command causes a problem:

  ```
  IQF> SELECT USER_ID, USER_GROUP, PROTECTION_GROUP-
  IQF> FROM USERS
  ```

  IQF reads `PROTECTION_GROUPFROM` as a column name. Correct this by inserting a space either before the hyphen or before `FROM` on the second line. Extra spaces do not affect the syntax.

- A keyword follows a command name. Enter it as shown in the format definition. To view the format definition for a command, type `HELP` followed by the command name at the IQF prompt. For example,

  ```
  IQF> HELP SAVE
  ```

  For details, see “Using Online Help” on page 2-12.

- Use single quotes around blanks and special characters, including single quotes themselves, to preserve their meaning in the command line.

  For example, to enter `USER ID` as part of a command line, enclose `USER ID` in single quotes as follows:

  ```
  IQF> FORMAT COLUMN USER_ID HEADING 'USER ID'
  ```

  To indicate `USER'S ID`, enter:

  ```
  IQF> FORMAT COLUMN USER_ID HEADING 'USER''S ID'
  ```

- For some commands and keywords, you can use an abbreviation instead of the entire word. These abbreviations appear as capital letters in the format definition for each command. For example, the format definition for the `SAVE` command is:

  ```
  SAVe filename REPlace APPend
  ```

SAV, REP, and APP are abbreviations for SAVE, REPLACE, and APPEND. The abbreviations are the minimum characters that you must enter. In the following example both commands have the same result:

```
IQF> SAVE filename APPEND
IQF> SAV filename APP
```

Using Views in IQF

To process a query, IQF accesses views. A view is a temporary representation of data from one or more tables. It is not necessary to have a one-to-one correspondence between a table and a view, since a view can be derived from more than a single table.

Vault creates views of tables by selecting only pertinent fields from a single table or several related tables. View definitions include descriptive column names. Sensitive fields, such as passwords, are excluded from view definitions.

Tables and Views Available to IQF

IQF can query any SQL table or view that has SQL `SELECT` privileges granted to PUBLIC, for example, the RDBMS data dictionary tables.

Your Vault System Administrator can make other tables and views—those that are not accessible to PUBLIC—available to IQF by granting `SELECT` privilege on the table or view to PDMQF, the SQL user ID used by IQF.

Please note:

- Refer to the documentation specific to your relational database management system for more information on SQL commands or on granting `SELECT` privileges.
- IQF uses only the created views and not the actual database tables. Vault does this with a separate RDBMS user ID that has read-only authority for views of the database tables. IQF uses the RDBMS user ID to access views. Consequently, you can access only those views authorized by this user ID.

You can create additional views of database tables and give read (`SELECT`) privileges for those views to the separate RDBMS user ID. For details on creating new views, see Appendix A, “Defining New Views for IQF Use.”

Please note: Complete information about the name, purpose, width, and data type of every column in all control tables and views is described in the *Vault Database Tables* manual.

The following table lists all the views provided with Vault:

Table 2-2 List of Views with Descriptions

View Name	Description
ACTIVE_XCTNS	Lists each transaction currently being processed.
AM_ATTRIBUTES	Lists names, data validation rule names, and descriptions of user-defined attributes.
AM_CLASSRULES	Lists names of simple rules used during installation.
AM_DATATYPES	Lists names of data types used during installation.
AM_FILEATTRIBUTES	Lists attributes and attribute data of Vault-controlled files.
AM_MEMBERATTRS	Lists members of a user-defined attributes set, whether the attribute is required or not, and the default attribute data.
AM_MEMBERSETS	Lists members of user-defined sets.
AM_PARTATTRIBUTES	Lists attributes and attribute data of Vault-controlled parts.
AM_RULES	Lists fragments of an SQL WHERE clause (keyword, operator, and test value) evaluated by the CLASSIFY function and the name of the user-defined attributes set used when the rule yields a TRUE state.
AM_SETS	Lists names and descriptions of user-defined sets.
APPL_ATTRIBUTES	Lists attributes used by EPD.Connect and other applications that use Rulebases.
APPL_DEFINITION	Lists attribute-related information for an application. This view is used by EPD.Connect.
APPL_ENVIRONMENT	Lists information related to application environments.
ARCHIVED_FILES	Lists information about archived Vault-controlled files.
ATTRDEFS	Lists information about binder attributes.
ATTRGRPS	Lists information about binder attribute groups.
AUTHORITY_GROUPS	Lists all authority levels and associated authority group names.
BACKUP_DIRECTORY	Lists names of Vault-controlled files copied to tape by the INCREMENTAL BACKUP command.
BATCH_PROCESSES	Lists completed batch-processing requests.
BATCH_REQUESTS	Lists all deferred requests requested by the user.
CA_FILEATTRIBUTES	Lists information about stored files and their associated attributes.
CA_FILE_DIRECTORY	Lists information about stored files.
CA_PARTATTRIBUTES	Lists information about stored parts and their associated attributes.
CA_PART_DIRECTORY	Lists information about stored parts.
COMMANDS	Lists all commands and their descriptions.
COMMAND_LISTS	Lists commands, their descriptions, and associated command list names.
COMMAND_TRIGGER	Lists commands that support triggers with the currently specified options.
CONFIGURATION	Lists configuration data and other information about the Vault.
CONNECT_VAULT	Provides a common method for accessing objects.

Table 2-2 List of Views with Descriptions

View Name	Description
DOM_IN_REVIEW	Lists the names of all Vault-controlled files currently being reviewed.
DOM_MESSAGE_KEY	Lists cross-reference information between user IDs and the messages they issue.
DOM_MESSAGE_TEXT	Lists the text of all the review-related messages issued to users.
DOM_REVIEWERS	Lists the ID of the message issued for each item being reviewed.
ENUMSETS	Used by binders to determine whether values of enumerated binder attributes are valid.
FILES_IN_REVIEW	Lists the names of all Vault-controlled files, initiated by a particular review request, that are currently in review.
FILE_DIRECTORY	Lists the names of Vault-controlled files.
FILE_SETS	Lists the names of file sets.
HEADING_DATA	Lists heading label data for the Data Browser.
IN_REVIEW	Lists names and requester information of Vault-controlled items currently being reviewed.
IN_REVIEW_VOTES	Lists names and votes of Vault-controlled items currently being reviewed.
JOBTYPES	Provides a method for some Optegra applications, such as Locator/PC, to define and maintain user roles, or “jobtypes”, that correspond to Vault Authority Groups.
LIST_TRFR_HEADINGS	Lists user preferences on how information in the EPD.Connect Data Browser must be displayed to the user.
LOG_DATA	Lists processing information about actions being performed by commands, such as security violations, successful processing notifications, and system errors.
MARKED_FOR_RESTORE	Lists all archived Vault-controlled files marked for restoration by the MARKR command.
MEMBER_SETS	Lists file sets in which a particular set member is referenced.
MESSAGES	Lists all users and messages issued.
OBJECT_DIRECTORY	Lists information about each Registered element in the Distributed environment.
PART_DIRECTORY	Lists names of Vault-controlled parts.
PART_FILES	Lists names of parts and their constituent files.
POOL_STATUS_CODES	Lists the status and description of all storage pools.
POOL_TYPES	Lists names and types of storage pools that are classified by a user-defined label.
PROJECTS	Lists the names and descriptions of all projects.
PROJECT_USERS	Lists all projects and users associated with each project.
PSN_MESSAGES	Lists file name and attribute information of Vault files.
PSN_REVIEWS	Lists information of files in review.
REFERENCES	Lists information of binder type definitions (typedefs) and objects.
REJECT_HISTORY	Lists information about attempted status changes of Vault-controlled files that did not occur.
REJECT_REVIEW_HIST	Lists detailed information about attempted status changes of Vault-controlled files that did not occur.
RELATIONS	Describes the relationships that can exist between binders and any binders, files, or parts.

Table 2-2 List of Views with Descriptions

View Name	Description
REVISION_CODES	Lists valid character revision codes of Vault-controlled files.
REVISION_HISTORY	Lists information about when the Vault-controlled file was assigned its released status code.
SET_MEMBERS	Lists set members contained within a given file set, commonly referred to as a single-level explosion.
SIGNED_OUT_FILES	Lists names of files currently signed out for modification.
SIGNED_OUT_GROUPS	Lists names of file sets and parts within file sets currently signed out for modification.
SIGNED_OUT_PARTS	Lists names of parts currently signed out for modification.
STANDALONE_FILES	Lists information on files not associated with a part.
STATUS_APPROVALS	Lists, for each project and status code, control information used when the Release/Revision Control process is used to change the status of Vault-controlled files.
STATUS_CODES	Lists status codes for both public and project files and authorization levels needed by users to access files of particular status codes.
STATUS_HISTORY	Lists information about each status code change that occurred for every Vault-controlled file.
STATUS_NOTIF_LISTS	Lists, for each project and status code, the notification list used to notify users when a review is initiated and when it is successfully completed.
STATUS_NOTIF_USERS	Lists, for each project and status code, the users who receive notifications when a review is initiated and when it is successfully completed.
STATUS_REVIEWERS	Lists, for each project and status code, the users who review and vote during the review process.
STATUS_REVIEW_HIST	Lists detailed information about successful Vault-controlled file status changes.
STORAGE_POOLS	Lists status and statistical information about each Vault-controlled storage area.
TABLE_DIRECTORY	Lists names of all Vault-controlled and user-defined tables.
TABLE_GRANTS	Lists, for each user-defined table, grants issued by Vault to the users.
TAPE_ALIASES	Lists, for each primary tape created by a command that specified an alternate or second copy tape, identifiers of the secondary tape.
UNIVERSAL_BACKUPS	Lists all current database backups performed by the UBKUP command for each storage pool.
USERS	Lists general information about each user.
USER_AUTHORITY	Lists general authorization information about each user in a project.
USER_COMMANDS	Lists commands that a user can perform on a project.
USER_LISTS	Lists names of all user lists.
USER_LIST_MEMBERS	Lists users, their occurrence on each user list, and names of other user lists.
VAULT_CONFIG	Used by Locator/PC and EPD.Connect to query the current Vault about its participation in the Distributed Vault environment.

Understanding IQF Messages

IQF messages appear when you enter:

- A non-IQF command
- Incorrect information with an IQF command

IQF messages also inform you why you cannot execute a certain command. For example, if you enter the `RUN` command and no command is in the buffer, IQF displays a message as follows:

```
IQF> RUN
CQFRUN108W There is no command currently active.
IQF>
```

Using Online Help

To view the online help, type `HELP` at the IQF prompt. IQF displays a list of IQF commands with a brief description of each command starting on the next line.

To get help for a particular command, type `HELP` followed by the name of the command at the prompt. For example:

```
IQF> HELP SAVE
```

```
SAVe filename [{ REPlace, APPend }]
```

filename — The complete name of a local file that will contain the current query result. Include as many as 80 alphanumeric characters. The name must follow the local file naming conventions.

`REPLace` — Replace the contents of the local file with the current query or group of IQF commands.

`APPend` — Append the current query (or group of IQF commands) to the contents of the local file.

Please note: IQF online help does not include comprehensive help for the `SELECT` command. For specific information about the `SELECT` command, refer to the documentation specific to your relational database management system.

Using Cases in IQF

Use any combination of letters, uppercase and lowercase, when entering IQF commands. However, to use the `SELECT` command, enter uppercase and lowercase letters to match the data stored in the tables.

Case Sensitivity and the Query Result

When you execute the `SELECT` command with a keyword such as `ORDER BY` on an RDBMS that distinguishes between uppercase and lowercase letters, the RDBMS displays the data in the order determined by the collating sequence.

Most operating systems use the ASCII character collating sequence.

Please note: For information on the character collating sequence used by your operating system, see the documentation specific to your operating system.

ASCII Character Collating Sequence

The ASCII character collating sequence in ascending order is as follows:

1. Space — blank
2. Special characters
3. Numbers — 0 through 9
4. More special characters
5. Uppercase letters — A to Z
6. More special characters
7. Lowercase letters — a to z
8. More special characters

The following are the general rules of the ASCII character collating sequence:

- Numbers before all letters
- Uppercase letters before lowercase letters
- Special characters almost anywhere

Chapter 3

Creating and Using Queries in IQF

This chapter describes the various command forms used to query IQF.

- Defining a Query Using the SELECT Command
- Executing the Current Query
- Displaying the Current Query
- Saving the Current Query
- Executing and Saving Multiple Commands in IQF
- Using IQF Predefined Queries
- Using IQF Command Files

Defining a Query Using the SELECT Command

To view specific information in the Vault control tables, execute a query in IQF. A query is used to specify the exact information you want to view.

The SELECT command is used to form a query. SELECT is a standard SQL command with many keywords that allow you to retrieve only the information you need. The following is a sample of the type of information you can specify with the SELECT command.

- View names.

  For details on the views used by IQF to display data, refer to "Using Views in IQF" on page 2-7.
- Column names.
- Columns to exclude.
- Order of columns.
- Order of rows.
- Conditions for displaying or hiding information.

The SELECT command executed most recently is called the current query. This could be either a SELECT command you entered or a START command you executed for a predefined query. For details on predefined queries, refer to "Using IQF Predefined Queries" on page 3-12.

Please note: When you enter IQF, there is no current query until you execute a SELECT command.

Format

```
SELECT columnname,. . . -
FROM viewname -
WHERE condition
```

Please note: A hyphen is used to continue a query on the next command prompt line, as described in "Conventions in IQF Commands" on page 2-5.

columnname The name of one or more columns to display. Enter an asterisk (*) to select all columns. Separate column names with a comma.

viewname The name of one or more views that contain the columns specified in the SELECT clause.

condition A condition for determining whether or not to select certain rows in the specified columns.

The format described in this section is not the complete format for the SELECT command. There are many more keywords that can be included. Consult the documentation specific to your RDBMS for complete information about the SELECT command.

Example

An example of the SELECT command follows:

```
IQF> SELECT USER_ID, LAST_NAME, PROTECTION_GROUP, USER_GROUP, -
IQF> UPDATE_DATE FROM USERS
```

The following is an example of a query using the WHERE option of the SELECT command:

```
IQF> SELECT USER_ID, LAST_NAME, PROTECTION_GROUP, -
IQF> USER_GROUP, UPDATE_DATE FROM USERS -
IQF> WHERE PROTECTION_GROUP = 'A3C2'
```

To view the results of the query, use either the FILE or PRINT command. For details on these commands, refer to Chapter 5, “Filing and Printing a Query Result.”

Using Placeholders and Parameters with a SELECT Command

A placeholder is a symbol in a SELECT command that is replaced with a value that you supply when you execute the query. A placeholder can be either a numeric or an alphabetic string, of either numbers or letters, preceded by the ampersand (&) symbol. When IQF encounters a placeholder in a SELECT command, it replaces it with the corresponding parameter in the START command.

Warning

Do not mix numeric and alphanumeric placeholders in a string. If you use numeric placeholders, the numbers in the string must be consecutive.

A SELECT command can have as many as nine placeholders. It can refer to each placeholder any number of times in any order. The number of parameters in the

START command must exactly match the number of placeholders in the SELECT command.

Example

```
IQF> SELECT * FROM FILE_DIRECTORY -
IQF> WHERE CLASS='&1' AND OWNER = '&2' -
IQF> ORDER BY FILE_REVISION,FILE_NAME
IQF> START (PRO ADMIN)
```

Here &1 and &2 are the placeholders while PRO and ADMIN are the parameters supplied. The resulting report is as follows:

```
FILE_NAME   FILE_REVISION     CLASS       OWNER       STATUS
---------   -------------     -----       -----       ------

PS4.&PS          1             PRO        ADMIN         RL
PS4.&PS          2             PRO        ADMIN         IW
```

Use the FILE or PRINT command to view the results of the query. For details on these commands, refer to Chapter 5, “Filing and Printing a Query Result.”

Executing the Current Query

To execute a query, use either of the following commands:

- The RUN command, to execute the current query when it has no placeholders.
- The START command, to execute the current query when it has placeholders.

Executing a Query without Placeholders

Use the RUN command to execute the current query when the query does not contain placeholders. This command is useful to execute a query that you did not explicitly save.

Please note:

- When you execute the current query with the RUN command, the query result reflects any format changes you have made since you last executed it.
- If a SELECT command has not been previously executed, the RUN command does nothing.

Format

```
RUN
```

Example

Consider the following query:

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, -
IQF> PROTECTION_GROUP, UPDATE_DATE FROM USERS
```

To execute this query use the RUN command as follows:

```
IQF> RUN
SELECT USER_ID, LAST_NAME, USER_GROUP, PROTECTION_GROUP,
UPDATE_DATE FROM USERS
```

IQF displays the SELECT command, unless ECHO is set to OFF, after you execute the RUN command. For details on ECHO options, refer to “Displaying Queries as They are Executed” on page 4-19.

Use the FILE or PRINT command to view the results of the query. For details on these commands, refer to Chapter 5, “Filing and Printing a Query Result.”

Executing a Query with Placeholders

Use the START command to execute the current query when it includes placeholders. IQF replaces the placeholders with the parameters you supply with the START command and then executes the query. The number of parameters must exactly match the number of placeholders.

If you do not supply parameters in the START command for the placeholders in the SELECT command, you get an error message.

Please note:

- IQF displays an error message if there is no current query.
- When you execute the current query again with the START command, the query reflects any format changes you have made since you last executed it.
- When IQF completes execution of the START command, the last SELECT command becomes the current query.

Format

STArt (parameter list)

parameter list — A list of parameters, separated by blanks, that are substituted for the placeholders in the SELECT command. Enclose the list in parentheses. Enclose a parameter that includes a blank in single quotes.

Example

Consider the following query:

```
IQF> SELECT USER_ID, &1, &2, PROTECTION_GROUP -
IQF> FROM USERS WHERE PROTECTION_GROUP = &3
```

To execute this query, specify three parameters with the START command as follows:

```
IQF> START (LAST_NAME USER_GROUP E23B)
```

The query is processed as though it were:

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, PROTECTION_GROUP -
IQF> FROM USERS WHERE PROTECTION_GROUP = E23B
```

Displaying the Current Query

Use the LIST command to display the current query. IQF displays a message when there is no current query.

If the current query has placeholders and you have executed a START command that replaces the placeholders with values, IQF lists the current query with those values. If you use the PLACEHOLDER option, IQF displays the current query with the placeholders instead of the values you supplied.

Format

```
LIST [PLACEHOLDER]
```

PLACEHOLDER	Displays the location of any placeholders (&1, &2, etc.) within the current query.

Example

Consider the following query:

```
IQF> SELECT &1, &2 FROM USERS
```

To execute this query, enter a START command with two parameters:

```
IQF> START (USER_ID PROTECTION_GROUP)
CQFSTA150I The START command has completed.
```

To display the current query, use the LIST command:

```
IQF> LIST
SELECT USER_ID, PROTECTION_GROUP FROM USERS
```

To display the placeholders in the current query:

```
IQF> LIST PLACEHOLDER
SELECT &1, &2 FROM USERS
```

Saving the Current Query

Use the SAVE command to save the current query in a local file. Doing this allows you to execute the query at a later time using the START command.

To save the current query in a new file, enter:

```
IQF> SAVE filename
```

If the file already exists, specify the REPLACE or APPEND option of the command. Otherwise, IQF sends you an error message and prompts you to do it.

Please note: Use the SAVE command with the EDIT command to save multiple IQF commands in a local file. For details on the EDIT command, refer to "Executing and Saving Multiple Commands in IQF" on page 3-10.

Format

```
SAVe filename [{REPlace, APPend}]
```

filename The complete name of a local file that will contain the current query. Include as many as 80 alphanumeric characters. The name must follow the local file-naming conventions.

REPLACE Replaces the contents of the local file with the current query.

APPEND Appends the current query to the contents of the local file.

Example

Consider the following current query:

```
IQF> SELECT USER_ID, USER_GROUP FROM USERS
```

To save the query in a new file named USER.REPORT:

```
IQF> SAVE USER.REPORT
```

A file USER.REPORT is created, containing the following command:

```
SELECT USER_ID, USER_GROUP FROM USERS
```

To replace the commands in this file with new commands:

```
IQF> SELECT USER_ID, USER_GROUP, PROTECTION_GROUP, -
IQF> UPDATE_DATE FROM USERS
IQF> SAVE USER.REPORT REPLACE
```

The file USER.REPORT now contains:

```
SELECT USER_ID, USER_GROUP, PROTECTION_GROUP, -
UPDATE_DATE FROM USERS
```

To add another command to the file USER.REPORT:

```
IQF> SELECT USER_GROUP, PROTECTION_GROUP FROM USERS
IQF> SAVE USER.REPORT APPEND
```

Finally, USER.REPORT contains the following commands:

```
SELECT USER_ID, USER_GROUP, PROTECTION_GROUP, -
UPDATE_DATE FROM USERS
SELECT USER_GROUP, PROTECTION_GROUP FROM USERS
```

Executing and Saving Multiple Commands in IQF

Use the `EDIT` command to create and save a predefined query. The commands you type in an `EDIT` session are not processed until you execute the `START` or `SAVE` command. You can enter a maximum of 54 IQF commands.

Please note: You cannot store these commands in a file with the `EDIT` command. This is because the `SAVE` and `START` commands begin processing with the previous commands, while the `END`, `EXIT`, and `QUIT` commands end the `EDIT` session.

Starting an EDIT Session

To start an `EDIT` session, execute the `EDIT` command at the IQF prompt as follows:

```
IQF> EDIT
EDT>
```

Enter IQF commands at this prompt.

Ending an EDIT Session

End an `EDIT` session without saving the commands as follows:

```
EDT> END
IQF>
```

End an edit session and save the commands as follows:

```
EDT> SAVE filename
IQF>
```

Warning

Save or execute the commands entered during the EDIT session before you end the session, or you will lose them.

Example

A sample `EDIT` session follows:

```
IQF> EDIT
EDT> SET ECHO ON
EDT> SELECT USER_ID, PROJECT_ID, READ_AUTHORITY, -
EDT> WRITE_AUTHORITY FROM USER_AUTHORITY
EDT> FORMAT COLUMN USER_ID HEADING ’USER ID’
EDT> FORMAT COLUMN PROJECT_ID HEADING ’PROJECT ID’
EDT> FORMAT TTITLE ’USER AUTHORITIES’
EDT> FORMAT BTITLE ’COMPANY CONFIDENTIAL’
EDT> START
SET ECHO ON
SELECT USER_ID, PROJECT_ID, READ_AUTHORITY, WRITE_AUTHORITY FROM
USER_AUTHORITY
FORMAT COLUMN USER_ID HEADING 'USER ID'
FORMAT COLUMN PROJECT_ID HEADING 'PROJECT ID'
FORMAT TTITLE 'USER AUTHORITIES'
FORMAT BTITLE 'COMPANY CONFIDENTIAL'
CQFSTA150I The START command has completed.
EDT> SAVE USER.REPORT
IQF>
```

For details on the `FORMAT` command and its options, refer to Chapter 4, “Formatting a Query Result.”

Using IQF Predefined Queries

The Interactive Query Facility is provided with a number of command files containing queries, that is, `SELECT` commands and other IQF commands that generate typical reports. You can use these files as they are or edit them to create customized reports.

The IQF command files are stored in the $EPD_HOME`/data` directory. These files have a `.iqf` suffix. To view a list of these files, use the following command at a UNIX command prompt:

```
% ls -l $EPD_HOME/data/*.iqf
```

The following table lists the name and description of each predefined query:

Table 3-1 IQF Predefined Query Files

Name of Query	View Used	Place-holders	Query Function
`All_files.iqf`	`FILE_DIRECTORY`	-	Displays all files currently known to EDM.
`All_projects.iqf`	`PROJECTS`	-	Displays a list of all projects.
`File_set_members.iqf`	`MEMBER_SETS`	-	Displays all information about file set members.
`Files_Signed_out_by_user.iqf`	`FILE_DIRECTORY`	user_id	Displays all files signed out by the specified user.
`Files_in_review.iqf`	`FILES_IN_REVIEW`	-	Displays all files currently in review.
`Files_on_project.iqf`	`FILE_DIRECTORY`	project_id	Displays all files on a selected project.
`In_review_by_proj.iqf`	`FILES_IN_REVIEW`	project_id	Displays files currently in review on a selected project.
`Log_errors.iqf`	`LOG_DATA`	-	Displays errors and warnings in log table.
`Part_audit.iqf`	`LOG_DATA`	part_name	Displays audit messages generated for a part.
`Pool_summary.iqf`	`STORAGE_POOLS`	-	Displays storage pool summary.
`Project_messages.iqf`	`MESSAGES, PROJECT_USERS`	project_id	Displays a list of active messages on a project.
`Rejns_by_file.iqf`	`REJECT_REVIEW_HIST`	file_name	Displays review failures for a selected file.
`Rejns_by_project.iqf`	`REJECT_REVIEW_HIST`	project_id	Displays history of rejections for files on a selected project.
`Security_violations.iqf`	`LOG_DATA`	-	Displays security violations.

Table 3-1 IQF Predefined Query Files

Name of Query	View Used	Place-holders	Query Function
`Status_history.iqf`	`STATUS_REVIEW_HIST`	project_id	Displays all information about every project.
`Users_on_project.iqf`	`PROJECT_USERS`	project_id	Displays the list of users on a selected project.
`Vote_status.iqf`	`IN_REVIEW_VOTES`	-	Displays the vote status of all files currently in review.
`edmfadfilecustom.iqf`	`STANDALONE_FILES`	user_id	Displays files signed out by a user.
`edmfadfileprivate.iqf`	`STANDALONE_FILES`	user_id	Displays all private files for owner.
`edmfadfileproject.iqf`	`STANDALONE_FILES`	project_id	Displays all files for a given project.
`edmfadfilepublic.iqf`	`FILE_DIRECTORY`	-	Displays all public files.
`edmfadfilereview.iqf`	`GUI_REVW_MESSAGES`	user_id	Displays all files in review for a given user.
`edmfadfilesignout.iqf`	`SIGNED_OUT_FILES`	user_id	Displays all files signed out to a user.
`edmfadpartcustom.iqf`	`PART_DIRECTORY`	user_id	Displays all parts signed out to a user.
`edmfadpartprivate.iqf`	`PART_DIRECTORY`	user_id	Displays all parts owned by a user.
`edmfadpartproject.iqf`	`PART_DIRECTORY`	project_id	Displays all parts belonging to a project.
`edmfadpartpublic.iqf`	`PART_DIRECTORY`	-	Displays all public parts.
`edmfadpartreview.iqf`	`GUI_REVW_MESSAGES`	user_id	Displays all parts to be reviewed by a user.
`edmfadpartsignout.iqf`	`SIGNED_OUT_PARTS`	user_id	Displays all parts signed out to `user_id`.
`edmfadsetfull.iqf`	`FILE_SETS`	-	Displays all file sets.
`edmfadsetreview.iqf`	`GUI_REVW_MESSAGES`	user_id	Displays all file sets in review for a user.
`list_signouts.iqf`	`SIGNED_OUT_FILES`	-	Displays all signed out files.
`listasn.iqf`	`FILE_DIRECTORY`	-	Displays all information about every project.
`listauth.iqf`	`USER_AUTHORITY`	-	Lists the authority attributes for every Vault user.
`listcat.iqf`	`FILE_DIRECTORY`	catalogname	Lists the Vault files in a CADDS 5*i* catalog.
`listcl.iqf`	`USER_COMMANDS`	-	Lists the Vault command list for each user in the database.

Table 3-1 IQF Predefined Query Files

Name of Query	View Used	Place-holders	Query Function
`listdir.iqf`	`FILE_DIRECTORY`	-	Lists the entire Vault file directory.
`listf.iqf`	`FILE_DIRECTORY`	-	Displays all information about every file.
`listfull.iqf`	`FILE_DIRECTORY`	-	Displays all information about every project.
`listp.iqf`	`PROJECTS`	-	Displays all information about every project.
`listproj.iqf`	`FILE_DIRECTORY`	-	Displays information about a project where `class=PRO`.
`listpu.iqf`	`PROJECT_USERS`	-	Displays all users on every project, by project.
`lists.iqf`	`STATUS_CODES`	-	Displays all information about the various status codes used on each project.
`listu.iqf`	`USERS`	-	Displays all information about every user.
`listup.iqf`	`PROJECT_USERS`	project_id	Displays IDs and names of all users on specified project.
`storage_pools.iqf`	`STORAGE_POOLS`	-	Displays all storage pool information.

Displaying the Contents of a Predefined Query

Use the `TYPE` command to display the contents of a predefined query. IQF sends you a message if the file name is incorrect or if the file does not exist.

Please note:

- If the predefined query is not found in the current working directory, it is searched for in the $EPD_HOME`/data` directory.
- Use the `TYPE` command to display the contents of any text file, not just those containing IQF commands, but also, for example, the filed results of queries. For details on filing a query result, refer to “Filing the Results of a Query” on page 5-2.

Alternatively, to display the contents of a predefined query, use the text editor provided with your operating system. For example, use the following command at a UNIX command prompt:

```
% more $EPD_HOME/data/filename.iqf
```

Format

TYPE filename

filename The complete name of a local file that contains IQF commands. Include as many as 80 alphanumeric characters. The name must follow the local file-naming conventions.

Example

To display the contents of Users_on_project.iqf, a predefined query:

```
IQF> TYPE Users_on_project.iqf
/***************************************************************/
/* @(#)Users_on_project.iqf     1.3 94/04/14 PCI                /
/*                                                              /
/* Name :       Users_on_project.iqf                            /
/* Usage :      start Users_on_project.iqf (project_id)         /
/* Function :   Display List of Users on a Selected Project    */
/***************************************************************/
set echo off
format ttitle ‘List of Users on a Selected Project’
set echo off
select * from project_users -
 where project_id = ‘&Project’ -
 order by project_id,user_id;
break on PROJECT_ID;
run
```

To display the contents of a filed query result:

```
IQF> TYPE QUERY.RESULT
MEMBER_NAME       MEMBER_TYPE   MEMBER_REV      MEMBER_DESC
-----------       -----------   ----------      -----------
CAMUNUM.&DB       F             2               Assembly
CAMUNUM.BOMADRAW  P             2               Adraw
NONREVIS.DEFAULT  P             1               Adraw
```

Executing a Predefined Query

To execute a predefined query, use the START command with a predefined query file name.

If a SELECT command in the command file includes placeholders, supply the parameters. See “Applying Placeholders and Parameters of a Predefined Query” on page 3-16 for more information. If you enter an incorrect file name, IQF displays an error message.

Please note: To find out if the `SELECT` command in the command file has placeholders, enter a `TYPE` command and the qualified command file name.

Format

`STArt` filename `[(`parameter list`)]`

filename A name that can include as many as 80 alphanumeric characters. It must follow the local file-naming conventions. It is the name of a file that contains a `SELECT` command or any other IQF commands.

parameter list A list of parameters, separated by blanks, that are substituted for the placeholders in the query command. Enclose the list in parentheses. Enclose a parameter that includes a blank in single quotation marks (`'`). The first parameter replaces every occurrence of the first placeholder, &string1, the second parameter replaces every occurrence of the second placeholder, &string2, and so on.

Example

To execute the predefined query `All_files.iqf`, use the `START` command as follows:

```
IQF> START All_files.iqf
CQFSTA150I The START command has completed.
IQF>
```

To view the results of the query, use the `FILE` or `PRINT` command. For details on these commands, refer to Chapter 5, “Filing and Printing a Query Result.”

Applying Placeholders and Parameters of a Predefined Query

Some predefined queries contain placeholders. To use these queries, supply the parameters as you execute the `START` command. For details on placeholders and parameters, refer to “Using Placeholders and Parameters with a SELECT Command” on page 3-3.

Example

Consider the following predefined query (Users_on_project.iqf) that contains a placeholder:

```
IQF> TYPE Users_on_project.iqf
/***************************************************************/
/* @(#)Users_on_project.iqf     1.3 94/04/14 PCI                /
/*                                                              /
/* Name :       Users_on_project.iqf                             /
/* Usage :      start Users_on_project.iqf (project_id)         /
/* Function :   Display List of Users on a Selected Project     /
/***************************************************************/
set echo off
format ttitle ‘List of Users on a Selected Project’
set echo off
select * from project_users -
 where project_id = ’&Project’ -
 order by project_id,user_id;
break on PROJECT_ID;
run
```

Use the START command to execute the query as follows:

```
IQF> START Users_on_project.iqf (GATEWAY)
CQFSTA150I The START command has completed.
IQF>
```

GATEWAY is the project ID (parameter) that replaces the placeholder &Project in the query. The query is processed as follows:

```
select * from project_users where project_id = GATEWAY -
order by project_id,user_id
```

Use the FILE or PRINT command to view the results of the query. For details on these commands, refer to Chapter 5, “Filing and Printing a Query Result.”

Using IQF Command Files

An IQF command file is a local file containing one or more IQF commands. You can have any number of such files. Each file name must follow the naming conventions of the local operating system.

IQF command files include the following:

- Predefined queries provided with IQF.

  These are stored in the $EPD_HOME/data directory.
- Queries you create and save using the SAVE command.

  For details on the SAVE command, refer to "Saving the Current Query" on page 3-8.
- A group of IQF commands created in the EDIT session or the operating system editor.

  For details on the EDIT command, refer to "Executing and Saving Multiple Commands in IQF" on page 3-10.

A command file can include more than a single SELECT command or no SELECT commands at all. If a command file does not contain a SELECT command, the query that was the current query when you executed the command file remains the current query.

The result generated is ignored when you execute the file unless you execute the FILE or PRINT command immediately after executing the file. For details on these commands, refer to Chapter 5, "Filing and Printing a Query Result."

Executing IQF Command Files

Execute an IQF command file as you would a predefined query. For details on executing a predefined query, refer to the section "Using IQF Predefined Queries" on page 3-12.

Chapter 4

Formatting a Query Result

This chapter explains the commands used to format the result of a query. Use these commands to specify the arrangement of information in the query result.

- Defining the Format of a Query Result
- Setting Format Parameters for a Query Result
- Organizing the Rows of a Query Result

Defining the Format of a Query Result

Use the FORMAT command to specify the format of the result generated by the current query. You can format a query result as follows:

- Specify:
  - Top and bottom titles.
  - The number of decimal places in a numeric column.
  - The width of a column.
- Define text for a null value.
- Turn formatting for a column off and on.
- Exclude columns from the display.
- Include columns that were previously excluded.
- Calculate subtotals and totals for numeric columns.
- Turn formatting for the query result off and on.

Rules for using FORMAT commands are as follows:

- To format a query result, execute a query first. FORMAT commands apply only to the current query.
- The FORMAT command has many options. Specify only a single option each time you execute the FORMAT command.
- After you execute a FORMAT command, execute the RUN command to effect the formatting.
- After executing the RUN command, file or print the query result to view the changes. For details on filing and printing a query result, refer to Chapter 5, “Filing and Printing a Query Result.”

Options defined using the FORMAT command remain in effect until you:

- Execute another SELECT command where at least one column is different.
- Execute another FORMAT command to change the current settings and execute the RUN command.
- Turn off all the current settings. See “Turning the Format Display Off and On” on page 4-15 for more information.
- Turn off the settings for an option.
- Exit IQF. See “Entering and Exiting IQF” on page 2-2 for more information.

Specifying Top and Bottom Titles

To specify titles for the top and bottom of every page of a query result, use the `TTITLE` and `BTITLE` options respectively.

Format

```
FORMat TTITle {'text', ON, OFF}

FORMat BTITle {'text', ON, OFF}
```

text — Defines the actual top or bottom title. A title can contain a maximum of 80 characters. IQF truncates the title if you use more than 80 characters. If the text contains a blank, enclose the text in single quotation marks (').

The top and bottom titles can each span one or two lines. Separate one line of the title from another with a vertical bar (|). Do not insert a space before or after the vertical bar. If the title is on two lines, the total number of characters in both lines must be less than 80 characters.

`ON` — Causes the specified top or bottom title to appear in the query result. This is the default.

`OFF` — Omits the specified top or bottom title from the query result.

Please note: When executed, the top and bottom title settings are active for the duration of the IQF session (unless specifically reset using the `OFF` option).

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, -
IQF> PROTECTION_GROUP, UPDATE_DATE FROM USERS
IQF> RUN
```

The query result is as follows:

```
USER_ID  LAST_NAME  USER_GROUP  PROTECTION_GROUP  UPDATE_DATE
-------- ---------- ----------  ----------------  -----------
EDMADMIN BROWN      ME          F23B              880425
AMELIA   EARHART    EE          E23B              880523
ORVILLE  WRIGHT     EE          DB12              880704
```

Specify top and bottom titles as follows:

```
IQF> FORMAT TTITLE 'USER INFORMATION|TEST GROUP'
IQF> FORMAT BTITLE 'COMPANY CONFIDENTIAL'
IQF> RUN
```

The query result is as follows when you specify top and bottom titles for the current query:

```
                                           USER INFORMATION
                                           PAGE   1
                                           TEST GROUP

USER_ID  LAST_NAME  USER_GROUP  PROTECTION_GROUP  UPDATE_DATE
-------- ---------- ----------- ----------------  -----------
EDMADMIN BROWN      ME          F23B              880425
AMELIA   EARHART    EE          E23B              880523
ORVILLE  WRIGHT     EE          DB12              880704
                                         COMPANY CONFIDENTIAL
```

Turn off the top title for the current query as follows:

```
IQF> FORMAT TTITLE OFF
IQF> RUN
```

The query result is as follows when you turn off the top title for the current query:

```
USER_ID  LAST_NAME  USER_GROUP  PROTECTION_GROUP  UPDATE_DATE
-------- ---------- ----------- ----------------  -----------
EDMADMIN BROWN      ME          F23B              880425
AMELIA   EARHART    EE          E23B              880523
ORVILLE  WRIGHT     EE          DB12              880704
                                         COMPANY CONFIDENTIAL
```

Specifying the Heading for a Column

Use the `COLUMN HEADING` option of the `FORMAT` command to define the heading for a particular column.

Format

`FORMat COLumn` column_name `HEADing` '*text*'

column_name The name of the column for which you want to define the heading.

text

The heading you want for the specified column. If the text contains a blank, enclose the text in single quotation marks ('). The maximum size of a column heading is the maximum number of characters allowed for the column. If you exceed that number, IQF truncates the heading. Enclose a column name in single quotes if the name contains a space. For an apostrophe in a name, use an extra single quote (for example, Fred’s Garage, would be `'Fred''s Garage'`).The column heading can span one or two lines. Separate the first line of the heading from the second with a vertical bar (|). Do not insert a space before or after the vertical bar. If the title is on two lines, the total number of characters in each line must be less than the maximum width of the column.

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, -
IQF> PROTECTION_GROUP, UPDATE_DATE FROM USERS
IQF> RUN
```

The query result is as follows:

```
USER_ID  LAST_NAME  USER_GROUP  PROTECTION_GROUP  UPDATE_DATE
-------- ---------- ----------- ----------------  -----------
EDMADMIN BROWN      ME          F23B              880425
AMELIA   EARHART    EE          E23B              880523
ORVILLE  WRIGHT     EE          DB12              880704
```

Format the heading of the `USER_ID` column as follows:

```
IQF> FORMAT COLUMN USER_ID HEADING ’USER ID’
IQF> RUN
```

The query result is as follows when you format the heading of the `USER_ID` column:

```
USER ID LAST_NAME USER_GROUP PROTECTION_GROUP UPDATE_DATE
------- --------- ---------- ---------------- -----------
EDMADMIN BROWN      ME          F23B              880425
AMELIA   EARHART    EE          E23B              880523
ORVILLE  WRIGHT     EE          DB12              880704
```

Format the heading of the `LAST_NAME` column as follows:

```
IQF> FORMAT COLUMN LAST_NAME HEADING ’LAST|NAME’
IQF> RUN
```

The query result is as follows when you format the heading of the LAST_NAME column:

```
USER ID  LAST       USER_GROUP  PROTECTION_GROUP  UPDATE_DATE
         NAME
-------- ---------- ----------- ----------------  -----------
EDMADMIN BROWN      ME          F23B              880425
AMELIA   EARHART    EE          E23B              880523
ORVILLE  WRIGHT     EE          DB12              880704
```

Specifying the Number of Decimal Places in a Column

Use the COLUMN DPLACES option of the FORMAT command to specify the number of decimal places that should appear in a numeric column.

Format

```
FORMat COLumn column_name DPLACES n
```

column_name The name of the column you want to format.

n Specifies the number of decimal places to be displayed/printed for a numeric column. Rounding is not performed. This value must be a positive integer from 1-10 (inclusive) that is less than the column width. The default is all the decimal places for the value being displayed or written.

Example

```
IQF> SELECT POOL_NAME, NO_FILES, POOL_SIZE, SIZE_USED, SIZE_FREE -
IQF> FROM STORAGE_POOLS ORDER BY POOL_NAME,POOL_SIZE
IQF> RUN
```

The query result is as follows:

```
POOL_NAME   NO_FILES    POOL_SIZE   SIZE_USED   SIZE_FREE
---------   --------    ---------   ---------   ---------
POOL1       1,083       86,761      70,700      16,061
POOL2       437         86,761      44,595      42,166
```

Specify two decimal places for the SIZE_USED column as follows:

```
IQF> FORMAT COLUMN SIZE_USED DPLACES 2
IQF> RUN
```

The query result is as follows when you specify two decimal places for the `SIZE_USED` column:

```
POOL_NAME    NO_FILES   POOL_SIZE   SIZE_USED   SIZE_FREE
---------    --------   ---------   ---------   ---------

POOL1        1,083      86,761      70,700.00   16,061
POOL2        437        86,761      44,595.00   42,166
```

Specifying the Width of a Column

Use the `COLUMN WIDTH` option of the `FORMAT` command to specify the width of a particular column in terms of the number of characters or digits it can contain.

Format

`FORMat COLumn` column_name `WIDth` n

column_name — The name of the column for which you want to define the width.

n — Specifies the total display width of the column. For character columns, n characters to the extreme left are displayed. For numeric columns, the n significant digits to the extreme left are displayed. This includes the sign and decimal point if they exist. This value must be a positive integer between 1 and 254, inclusive of 1 and 254. The default is the larger of the column name and the full width of the column. For character data, if the actual data to be displayed in the column is longer than the specified width, IQF truncates the data to fit in the column. If a column is numeric, however, and the specified width is less than the number of digits needed to display the number, IQF replaces the entire field with asterisks (********).

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP FROM USERS
IQF> RUN
```

The query result is as follows:

```
USER_ID     LAST_NAME   USER_GROUP
-------     ---------   ----------
EDMADMIN    BROWN       EE
ORVILLE     WRIGHT      EE
```

Specify the width of the column USER_ID as follows:

```
IQF> FORMAT COLUMN USER_ID WIDTH 25
IQF> RUN
```

The query result is as follows when you specify the width of the USER_ID column:

```
USER_ID                        LAST_NAME         USER_GROUP
-------                        ---------         ----------
EDMADMINISTRATOR               BROWN             EE
ORVILLE                        WRIGHT            EE
```

Defining Text for Null Values in a Column

Use the COLUMN NULL option of the FORMAT command to specify the text to be displayed when there is a null value in a column of the query result.

A null value is an empty field and the display shows a blank. When there is no value available to display in the column, the resulting blank in the query result is called a null value.

Please note: To specify text for all the null values in every query you execute, see “Defining Text for Null Values” on page 4-22.

Format

FORMat COLumn column_name NULL 'text'

column_name The name of the column for which you want to define a null value.

text Specifies the text to be displayed for a null value. Text is a character string of as many as 254 characters. This text replaces null values in the query result. The default is a string of blanks.

If the text contains a blank, enclose it in single quotation marks. If the length of the text is greater than the width of the column, IQF truncates the text with no warning.

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, DESCRIPTION -
IQF> FROM USERS ORDER BY USER_ID
IQF> RUN
```

The query result is as follows:

```
USER_ID     LAST_NAME   USER_GROUP   DESCRIPTION
-------     ---------   ----------   -----------
AMELIA      EARHARE     EET          PROJECT LEADER
BERYL       MARKUM      ME
CHARLES     LINDBERGH   ME           OPERATOR
ERICA       JONGEE
ORVILLE     WRIGHTEE    CARGO        PROJECT LEADER
EDMADMIN    BROWN       ME           SYSTEM ADMINISTRATOR
```

Specify a null value for the `DESCRIPTION` column as follows:

```
IQF> FORMAT COLUMN DESCRIPTION NULL ********************
IQF> RUN
```

The query result is as follows when you specify a null value for the `DESCRIPTION` column:

```
USER_ID     LAST_NAME   USER_GROUP   DESCRIPTION
-------     ---------   ----------   -----------
AMELIA      EARHARE     EET          PROJECT LEADER
BERYL       MARKUM      ME           ********************
CHARLES     LINDBERGH   ME           OPERATOR
ERICA       JONGEE                   ********************
ORVILLE     WRIGHTEE    CARGO        PROJECT LEADER
EDMADMIN    BROWN       ME           SYSTEM ADMINISTRATOR
```

Turning Formatting for a Column Off and On

Use the `COLUMN OFF` and `COLUMN ON` options of the `FORMAT` command to turn formatting off and on for a particular column.

Format

`FORMat COLumn` column_name `{OFF, ON}`

column_name The name of the column for which you want to turn formatting off or on.

`OFF` Turns off the previously defined heading and null text strings. It also returns the settings for decimal places and width to their original values.

`ON` Turns back on the previously defined heading, null text strings, and settings for decimal places and width. The default is ON.

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP FROM USERS
IQF> FORMAT COLUMN USER_ID WIDTH 30
IQF> FORMAT COLUMN LAST_NAME HEADING 'LAST|NAME'
IQF> RUN
```

The query result is as follows:

```
USER_ID                        LAST NAME         USER_GROUP
-------                        ---------         ----------
EDMADMINISTRATOR               BROWN             ME
AMELIA                         EARHART           EE
ORVILLE                        WRIGHT            EE
```

Turn off the formatting for the USER_ID and LAST_NAME columns as follows:

```
IQF> FORMAT COLUMN USER_ID OFF
IQF> FORMAT COLUMN LAST_NAME OFF
IQF> RUN
```

The query result is as follows when you turn off the formatting for the USER_ID and LAST_NAME columns:

```
USER_ID     LAST_NAME    USER_GROUP
-------     ---------    ----------
EDMADMIN    BROWN        ME
AMELIA      EARHART      EE
ORVILLE     WRIGHT       EE
```

Excluding Columns from the Display

Use the EXCLUDE option of the FORMAT command to exclude certain columns from the display of the query result.

Format

```
FORMat EXCLude [ALL BUT] (column_list)
```

column_list A list of column names as they appear in the SELECT command. Except when used with the ALL BUT option, these are the columns that you do not want to appear in the display.When specifying more than a single column, separate the names with a blank and enclose the list in parentheses.

ALL BUT Indicates that you are listing the columns that you want to appear in the display. The display excludes all columns that do not appear in the column list. This command is equivalent to the INCLUDE ONLY command.

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, -
IQF> PROTECTION_GROUP, UPDATE_DATE FROM USERS
IQF> FORMAT EXCLUDE ALL BUT (USER_ID LAST_NAME USER_GROUP)
IQF> RUN
```

The query result is as follows:

```
USER_ID          LAST_NAME        USER_GROUP
-------          ---------        ----------
EDMADMIN         BROWN            ME
AMELIA           EARHART          EE
ORVILLE          WRIGHT           EE
```

Exclude the USER_GROUP column as follows:

```
IQF> FORMAT EXCLUDE USER_GROUP
IQF> RUN
```

The query result is as follows when you exclude the USER_GROUP column:

```
USER_ID          LAST_NAME
-------          ---------
EDMADMIN         BROWN
AMELIA           EARHART
ORVILLE          WRIGHT
```

Including Columns Previously Excluded from the Display

Use the INCLUDE option of the FORMAT command to include columns in the display that you previously excluded. This option reverses the effect of the EXCLUDE option. If you execute the INCLUDE option without specifying any column names, all columns that were previously excluded are now included in the display.

Format

```
FORMat INCLude [[ONLY](column_list)]
```

`ONLY` Indicates that you are listing the columns that you want to appear in the display. The display includes only the columns that appear in the list. This command is equivalent to the `EXCLUDE ALL BUT` command.

column_list A list of column names as they appear in the `SELECT` command. Except when used with the `ONLY` option, these are the columns that you want to appear in the display again. When specifying more than one column, separate the names with a blank and enclose the list in parentheses.

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, -
IQF> PROTECTION_GROUP, UPDATE_DATE FROM USERS
IQF> RUN
```

The query result is as follows:

```
USER_ID          LAST_NAME       USER_GROUP    PROTECTION_GROUP
-------          ---------       ----------    ----------------
EDMADMIN         BROWN           ME            F23B
AMELIA           EARHART         EE            E23B
ORVILLE          WRIGHT          EE            DB12
```

Exclude the `PROTECTION_GROUP` column from the query result as follows:

```
IQF> FORMAT EXCLUDE ALL BUT (USER_ID LAST_NAME USER_GROUP)
IQF> RUN
```

The query result is as follows when you exclude the `PROTECTION_GROUP` column from the query result:

```
USER_ID        LAST_NAME      USER_GROUP
-------        ---------      ----------
EDMADMIN       BROWN          ME
AMELIA         EARHART        EE
ORVILLE        WRIGHT         EE
```

Include the `PROTECTION_GROUP` column as follows:

```
IQF> FORMAT INCLUDE PROTECTION_GROUP
IQF> RUN
```

The query result is as follows when you include the PROTECTION_GROUP column:

```
USER_ID    LAST_NAME    USER_GROUP     PROTECTION_GROUP
-------    ---------    ----------     ----------------
EDMADMIN   BROWN        ME             F23B
AMELIA     EARHART      EE             E23B
ORVILLE    WRIGHT       EE             DB12
```

Exclude all except the USER_ID column as follows:

```
IQF> FORMAT INCLUDE ONLY USER_ID
IQF> RUN
```

The query result is as follows:

```
USER_ID
-------
EDMADMIN
AMELIA
ORVILLE
```

Calculating Totals and Subtotals for Columns

Use the TOTAL and SUBTOTAL options of the FORMAT command to calculate the total or subtotal for a specified numeric column.

These options are used with the BREAK ON column command to display totals and subtotals when the value of column changes. For details on the BREAK command, refer to “Organizing the Rows of a Query Result” on page 4-27.

Format

FORMat TOTal {[EXCEPT](column_list), ON, OFF}

FORMat SUBTotal {[EXCEPT](column_list), ON, OFF}

EXCEPT — Indicates that you are listing the columns for which you do not want to calculate totals or subtotals. The IQF calculates totals or subtotals for the numeric columns that are not listed.

column_list — A list of column names as they appear in the SELECT command. Except when used with the EXCEPT option, these are the columns for which you want to do calculations. When specifying more than one column, separate the names with a blank and enclose the list in parentheses. You can specify numeric columns only. The TOTAL and SUBTOTAL options do not work on character columns.

`ON` Turns on the calculation of a total or subtotal.

`OFF` Turns off the calculation of a total or subtotal.

Please note:

- When you specify the `SUBTOTAL` option, IQF also calculates a total unless you execute the `FORMAT TOTAL OFF` command.
- If the total or subtotal result is larger than the width of the column, IQF replaces the entire field with asterisks (********). Use the `COLUMN WIDTH` of the `FORMAT` command to change the width of a column. For details on formatting the width of a column, refer to “Specifying the Width of a Column” on page 4-7.

Example

```
IQF> SELECT DEPT_NUM, EMP_NUM, EMP_NAME, SALARY, COMM -
IQF> FROM EMP ORDER BY DEPTNO, EMPNO
IQF> RUN
```

Please note: `EMP` is not a view provided with IQF. For information on creating new views, refer to Appendix A, “Defining New Views for IQF Use.”

The query result is as follows:

```
DEPT_NUM     EMP_NUM    EMP_NAME   SALARY       COMM
--------     -------    --------   ------       ----
10           7839       WOOD       5,500.00
10           7934       MILLER       920.00
10           7999       BROWN        800.00
20           7369       COHEN        920.00
20           7566       JONES      3,123.75
20           7657       HILL       3,910.00
20           7876       ASCI         920.00
20           7902       FORD       3,450.00
30           7521       WARD       1,312.50       500.00
30           7654       MARTIN     1,312.50     1,400.00
30           7900       JAMES        920.00
30           7955       WILSON     2,000.00       500.00
```

Break the query result and add subtotals as follows:

```
IQF> BREAK ON DEPTNO SKIP 2
IQF> FORMAT SUBTOTAL (SALARY COMM)
IQF> RUN
```

The query result is as follows when you break the query result and add subtotals:

```
DEPT_NUM   EMP_NUM   EMP_NAME   SALARY      COMM
--------   -------   --------   ------      ------
10         7839      WOOD       5,500.00
           7934      MILLER       920.00
           7999      BROWN        800.00
*******    ****      ******     ---------   ------
                                7,220.00      0.00

20         7369      COHEN        920.00
           7566      JONES      3,123.75
           7657      HILL       3,910.00
           7876      ASCI         920.00
           7902      FORD       3,450.00
*******    ****      ****       ---------   ------
                               12,323.75      0.00

30         7521      WARD       1,312.50    500.00
           7654      MARTIN     1,312.50  1,400.00
           7900      JAMES        920.00
           7955      WILSON     2,000.00    500.00
*******    ****      ******     --------- ----------
                                5,545.00  2,400.00
*********************           --------  ----------
TOTAL                          25,088.75  2,400.00
```

Turning the Format Display Off and On

Use the `OFF` and `ON` options with the `FORMAT` command to turn the formatting for the query result off and on. To cancel all formatting done on the current query, use the `ERASE` option.

Format

```
FORMat {OFF, ON, ERASE}
```

`OFF` Turns off the display of all previously defined `FORMAT` settings.

`ON` Previously defined `FORMAT` settings appear in the display. This is the default.

`ERASE` Cancels all previously defined `FORMAT` options and resets the `FORMAT` settings back to their original values.

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, -
IQF> PROTECTION_GROUP, UPDATE_DATE FROM USERS
IQF> RUN
```

The query result is as follows:

```
USER_ID   LAST_NAME   USER_GROUP   PROTECTION_GROUP   UPDATE_DATE
-------   ----------  ----------   ----------------   -----------
EDMADMIN  BROWN        ME          F23B               880425
AMELIA    EARHART      EE          E23B               880523
ORVILLE   WRIGHT       EE          DB12               880704
```

Format the query result as follows:

```
IQF> FORMAT TTITLE 'USER INFORMATION'
IQF> FORMAT BTITLE 'COMPANY CONFIDENTIAL'
IQF> FORMAT COLUMN USER_ID HEADING 'USER ID'
IQF> FORMAT COLUMN LAST_NAME HEADING 'LAST|NAME'
IQF> FORMAT COLUMN USER_ID WIDTH 30
IQF> FORMAT INCLUDE ONLY (USER_ID LAST_NAME)
IQF> RUN
```

Then, the query result is as follows:

```
              USER INFORMATION
                      PAGE   1

USER_ID              LAST NAME
-------              ---------
EDMADMINISTRATOR     BROWN
AMELIA               EARHART
ORVILLE              WRIGHT

          COMPANY CONFIDENTIAL
```

Disable all the FORMAT settings as follows:

```
IQF> FORMAT OFF
IQF> RUN
```

Then, the query result is as follows:

```
USER_ID   LAST_NAME   USER_GROUP   PROTECTION_GROUP   UPDATE_DATE
-------   ----------  ----------   ----------------   -----------
EDMADMIN  BROWN        ME          F23B               880425
AMELIA    EARHART      EE          E23B               880523
ORVILLE   WRIGHT       EE          DB12               880704
```

Enable the previous FORMAT settings as follows:

```
IQF> FORMAT ON
IQF> RUN
```

Then, the query result is as follows:

```
               USER INFORMATION
                       PAGE   1

USER_ID              LAST NAME
-------              ---------
EDMADMINISTRATOR     BROWN
AMELIA               EARHART
ORVILLE              WRIGHT

           COMPANY CONFIDENTIAL
```

Setting Format Parameters for a Query Result

Use the `SET` command to define values for the following format parameters:

- Punctuation in decimal columns
- Display queries as they are executed
- Line width
- Number of blank lines at the top of every page
- Number of lines of data printed per page
- Total number of lines per page
- Text to be printed for null values
- Characters used to separate columns

Rules for using `SET` commands follow:

- You can define only a single parameter with a `SET` command.
- When you define a parameter, it is effective until you specifically reset the parameter or exit IQF.
- The results of changed parameters are reflected only in the file or in the printed query result. For details on filing and printing a query result, refer to Chapter 5, "Filing and Printing a Query Result."

Please note: Each parameter has a default setting. Execute the following command to display the default settings for your system:

```
IQF> SHOW ALL
```

For details on the `SHOW` command, refer to "Displaying the Value of Format Parameters" on page 4-25.

Defining Punctuation in Decimal Columns

The `SET DECIMAL` command allows you to define punctuation in decimal columns of the query result.

Format

```
SET DECimal /t/d/
```

t The thousands separator. You can enter a period, a comma, or a blank. The default is no separator for thousands.

d The decimal separator. You can enter a period or a comma. The default separator for decimals is a period.

Example

```
IQF> SET DECIMAL /,/./
```

This command causes the thousands separator to be a comma and the decimal separator to be a period. The number 10456.99 is displayed as 10,456.99.

The only valid combinations follow:

```
IQF> SET DECimal /./,/
IQF> SET DECimal /,/./
IQF> SET DECimal //,/
IQF> SET DECimal //./
```

Please note: Use the last two combinations to avoid specifying a thousands separator.

Displaying Queries as They are Executed

Use the `SET ECHO` command to specify whether or not to display queries as IQF executes them.

Format

```
SET ECHO {OFF, ON}
```

`OFF` Indicates that stored queries are not displayed before they are executed.

`ON` Indicates that stored queries are displayed before they are executed. This is the default.

Example

```
IQF> SET ECHO OFF
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, PROTECTION_GROUP, -
IQF> UPDATE_DATE FROM USERS
IQF> RUN
IQF>
IQF> SET ECHO ON
IQF> RUN
SELECT USER_ID, LAST_NAME, USER_GROUP, PROTECTION_GROUP,
UPDATE_DATE FROM USERS
IQF>
```

Use the `SET ECHO ON` command to view the parameters in a predefined query as you execute it. An example follows:

```
IQF> START listup.iqf (GATEWAY)
CQFSTA150I The START command has completed.
IQF> SET ECHO ON
IQF> RUN
SELECT * FROM PROJECT_USERS WHERE PROJECT_ID = 'GATEWAY' ORDER BY
PROJECT_ID, USER_ID
IQF>
```

For details on parameters in predefined queries, refer to “Applying Placeholders and Parameters of a Predefined Query” on page 3-16.

Defining the Number of Characters in a Line

Use the `SET LINESIZE` command to specify the maximum number of characters on a single line of the query result.

Format

```
SET LINesize n
```

n A three-digit positive integer. It specifies the number of characters that can be printed or filed per line. The default is 132 characters.

Example

```
IQF> SET LINESIZE 80
```

Each line in the query result contains a maximum of 80 characters.

Setting the Top Margin for Each Page

Use the `SET NEWPAGE` command to define the top margin for each page of the query result.

Format

```
SET NEWPage n
```

n A two-digit positive integer. It is the number of blank lines to be printed or filed at the top of every page. A value of 0 sends a form feed between pages and does not print any blank lines at the top of the page. The default is 1.

Example

```
IQF> SET NEWPAGE 3
```

This creates three blank lines at the top of each page of the query result.

Defining the Number of Data Lines in a Page

Use the `SET NOROWS` command to specify the maximum number of rows of data on each page of the query result. This does not include top and bottom titles or column headings.

Format

```
SET NOROWS n
```

n A two-digit positive integer. It is the number of lines of data to be printed or filed per page. The default value is 54.

Example

```
IQF> SET NOROWS 16
```

A maximum of 16 lines of data is printed or filed on each page of the query result.

Defining the Number of Lines in a Page

Use the `SET PAGESIZE` command to specify the total number of lines that you can include on each page of the query result. This number includes top and bottom titles and column headings.

Format

```
SET PAGESize n
```

n A two-digit positive integer. It specifies the total number of lines per page. The default is 66.

Example

```
IQF> SET PAGESIZE 30
```

A maximum of 30 lines is printed or filed on each page of the query result.

Defining Text for Null Values

Use the `SET NULL` command to specify the text for a null value in the query result.

Please note: This command achieves the same result as the `FORMAT` command described in the section “Defining Text for Null Values in a Column” on page 4-8.When specified, the `FORMAT` command overrides the `SET NULL` command.

A null value is an empty field and the display shows a blank. When there is no value available to display in the column, the resulting blank in the query result is called a null value.

Format

```
SET NULL 'text'
```

text Specifies the text to be displayed for a null value. Text is a character string of as many as 254 characters. This text replaces null values in the query result. The default is a string of blanks. If the text contains a blank, enclose it in single quotation marks. If the length of the text is greater than the width of the column, IQF truncates the text with no warning.

Example

```
IQF> SELECT FILE_NAME, PART_NUMBER, DESCRIPTION -
IQF> FROM FILE_DIRECTORY
IQF> WHERE FILE_NAME LIKE 'JET.ENGINE2.%'
IQF> ORDER BY FILE_NAME
IQF> FORMAT COLUMN FILE_NAME WIDTH 20
IQF> RUN
```

The result is as follows:

```
FILE_NAME              PART_NUMBER           DESCRIPTION
---------              -----------           -----------
JET.ENGINE2.&BCD       9010-AS1JET-NLM
JET.ENGINE2.&NFIG
JET.ENGINE2.&PD        9010-AS1JET-NLM       ENGINE #2
JET.ENGINE2.&PFIG
JET.ENGINE2.&SFIG
JET.ENGINE2.DOCUMENT   9010-AS1JET-NLM       ASSEMBLY INSTRUCTIONS
JET.ENGINE2.DWG1                             TOP VIEW - 1A-2B
JET.ENGINE2.DWG2                             TOP VIEW - 2B-3C
JET.ENGINE2.NOTES                            NOTES-IN-PROGRESS
JET.ENGINE2.SPECS      9010-AS1JET-NLM       INITIAL SPECIFICATION
```

Set 'Not Entered' as the value to replace `all blank fields` as follows:

```
IQF> SET NULL 'NOT ENTERED'
```

The result is as follows:

```
FILE_NAME              PART_NUMBER           DESCRIPTION
---------              -----------           -----------
JET.ENGINE2.&BCD       9010-AS1JET-NLM       NOT ENTERED
JET.ENGINE2.&NFIG                            NOT ENTERED
JET.ENGINE2.&PD        9010-AS1JET-NLM       ENGINE #2
JET.ENGINE2.&PFIG                            NOT ENTERED
JET.ENGINE2.&SFIG                            NOT ENTERED
JET.ENGINE2.DOCUMENT   9010-AS1JET-NLM       ASSEMBLY INSTRUCTIONS
JET.ENGINE2.DWG1                             TOP VIEW - 1A-2B
JET.ENGINE2.DWG2                             TOP VIEW - 2B-3C
JET.ENGINE2.NOTES                            NOTES-IN-PROGRESS
JET.ENGINE2.SPECS      9010-AS1JET-NLM       INITIAL SPECIFICATION
```

Defining Column Separators

Use the `SET SEPARATOR` command to define how to separate columns in the query result. You can use any character, including a blank.

Format

```
SET SEParator {n BLAnks, 'text'}
```

n A three-digit positive integer. Specify n `BLANKS` to set the number of blanks that appear between any two columns. The default is two blanks.

text A character string of as many as 10 characters. This text appears between any two columns. If the text contains a blank, enclose it in single quotation marks. If the length of the text is greater than the width of the column, IQF truncates the text with no warning.

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, -
IQF> PROTECTION_GROUP FROM USERS
IQF> RUN
```

The query result is as follows:

```
USER_ID       LAST_NAME        USER_GROUP
------------  ---------------  ----------

EDMADMIN   BROWN        ME
AMELIA     EARHART      EE
ORVILLE    WRIGHT       EE
```

Set a separator of six blanks between each column as follows:

```
IQF> SET SEPARATOR 6 BLANKS
IQF> RUN
```

The query result is as follows:

```
USER_ID           LAST_NAME            USER_GROUP
------------      ---------------      ----------

EDMADMIN          BROWN                ME
AMELIA            EARHART              EE
ORVILLE           WRIGHT               EE
```

Set pipe (|) as the separator between columns as follows:

```
IQF> SET SEPARATOR |
IQF> RUN
```

The query result is as follows:

```
USER_ID     |LAST_NAME      |USER_GROUP|
------------|---------------|----------|

EDMADMIN    |BROWN          |ME
AMELIA      |EARHART        |EE
ORVILLE     |WRIGHT         |EE
```

Displaying the Value of Format Parameters

Use the SHOW command to display the value of one or all format parameters.

Format

```
SHOW {ALL, BTITle, DECimal, ECHO, LINesize, NEWPage, NOROWS,
NULL, PAGESize, SEParator, TTITle}
```

ALL — Shows the current settings for all the parameters as listed.

BTITLE — Shows the current bottom title for query results.

DECIMAL — Shows the type of punctuation used when displaying a decimal column.

ECHO — Shows whether commands are displayed when executed from a command file.

LINESIZE — Shows the maximum number of characters that can be displayed or printed on a line.

`NEWPAGE` Shows the number of blank lines printed at the top of every page.

`NOROWS` Shows the maximum number of lines of data printed per page.

`NULL` Shows the text that is substituted for a null value.

`PAGESIZE` Shows the total number of lines per page.

`SEPARATOR` Shows the separation characters used between columns.

`TTITLE` Shows the current top title for query results.

Example

```
IQF> SHOW NEWPAGE
NEWPAGE    is   3
IQF> SHOW PAGESIZE
PAGESIZE   is  24
IQF> SHOW ALL
BTITLE     is  NULL
DECIMAL    is  /,/./
ECHO       is  ON
LINESIZE   is  79
NEWPAGE    is  1
NOROWS     is  54
NULL       is
SEPARATOR  is  2 BLANK(S)
PAGESIZE   is  66
TTITLE     is  NULL
IQF>
```

Organizing the Rows of a Query Result

Use the BREAK command to add and format breaks in a query result. These breaks help to prioritize information in a query result. This command applies only to the current query.

When you execute the BREAK command, the breaks are reflected in a file or a printed query result. For details on filing and printing a query result, refer to Chapter 5, "Filing and Printing a Query Result."

Adding Breaks in a Query Result

The BREAK ON command inserts a break in the query result in the following cases:

- When the value of the specified columns changes
- After every row
- After every page
- At the end of the query result

You can break several columns at a time.

Format

```
BREak ON {column, ROW, PAGe, REPORT}
```

column The name of a column in the current query. When the value of this column changes, IQF inserts a break in the output. If subtotals or totals are requested, they appear in the appropriate columns each time the value of column changes.

ROW Causes a break after every row of data is displayed. If subtotals or totals are requested, they appear in the appropriate columns after each row of data.

PAGE Causes a break at the end of each page. If subtotals or totals are requested, they appear in the appropriate columns at the bottom of each page of data.

REPORT Causes a break at the end of a report or query. If subtotals or totals are requested, they appear in the appropriate columns at the end of the report.

Except for the `BREAK ON` column command, the break settings remain in effect until you turn them off either by using the `BREAK OFF` command or by exiting IQF. IQF turns off the breaks specified for a column when you execute a new query.

Use the `BREAK ON` column command with the `FORMAT TOTAL` and `FORMAT SUBTOTAL` commands to display totals and subtotals whenever the value of the specified column changes.

Example

```
IQF> SELECT LOG_DATE, LOG_TIME, TASK_ID, USER_ID, NODE_NAME -
IQF> FROM LOG_DATA WHERE NODE_NAME = 'hathi' -
IQF> ORDER BY LOG_DATE,TASK_ID
IQF> BREAK ON LOG_DATE
IQF> RUN
```

The result is as follows:

```
LOG_DATE    LOG_TIME    TASK_ID    USER_ID     NODE_NAME
--------    --------    -------    -------     ---------

990827      103419      edmapl     HATHI       hathi
            103450      edmapl     HATHI       hathi

990908      153937      edmapl     HATHI       hathi
            153918      edmapl     HATHI       hathi

990910      153958      edmapl     EDMADMIN    hathi
            155026      edmapl     EDMADMIN    hathi
```

Formatting Breaks in a Query Result

After you specify where a break must occur, specify whether IQF must:

- Skip a certain number of lines at the break.
- Skip to the next page.
- Print or display the value in the column that causes the break only when the value changes.
- Print or display all values in the column that causes the break.

Format

```
BREak ON column [SKIp {n, PAGE}] [{NODUPlicates, DUPlicates}]
```

`SKIP n` A two-digit positive integer. It specifies the number of lines IQF must skip when there is a break. IQF skips the specified number of lines and then prints any applicable totals or subtotals. The default is 1. Applicable only when you specify column.

`SKIP PAGE` Specifies that IQF must skip to the next page when there is a break. IQF prints any applicable totals or subtotals at the bottom of the page before going to the new page. Applicable only when you specify column.

`NODUPLICATES` Specifies that the value in the column that causes the break must be displayed or printed only when that value changes. In other words, IQF displays or prints the value only in the first row in which the value occurs. This is the default. Applicable only when you specify column.

`DUPLICATES` Specifies that all values in the column that caused the break must be displayed or printed. Applicable only when you specify column.

Example

```
IQF> SELECT FILE_NAME, CLASS, OWNER, NUM_UNIT -
IQF> FROM ARCHIVED_FILES ORDER BY OWNER
IQF> FORMAT COLUMN FILE_NAME WIDTH 10
IQF> RUN
```

The query result is as follows:

```
FILE_NAME   CLASS       OWNER       NUM_UNIT
---------   -----       -----       --------

KTEST&14.&   PUB        PN001        0
KTEST&14_R   PUB        PD001        0
PARTS1.ASS   PUB        PD001        0
PARTS1.ASS   PUB        PD001        0
PARTS1.ASS   PUB        GATEWAY      0
PARTS1.ASS   PUB        GATEWAY      0
```

Add breaks in the query result as follows:

```
IQF> BREAK ON OWNER SKIP 2 DUPLICATES
IQF> RUN
```

The query result is as follows:

```
FILE_NAME    CLASS      OWNER      NUM_UNIT
---------    -----      -----      --------

KTEST&14.&    PUB       PN001      0

KTEST&14_R    PUB       PD001      0

PARTS1.ASS    PUB       PD001      0

PARTS1.ASS    PUB       PD001      0

PARTS1.ASS    PUB       PD001      0

PARTS1.ASS    PUB       PD001      0
```

Chapter 5

Filing and Printing a Query Result

This chapter describes the commands used to file and print the results of a query. To view the results of a query in IQF you must file or print a report.

- Filing the Results of a Query
- Printing the Results of a Query

Filing the Results of a Query

Use the FILE command to save the result of the current query in a file. IQF saves the query result in a file with the formatting you have specified or the default values. The only exception to this is when you specify the DATAONLY option of the FILE command.

The DATAONLY option copies the data from the query result to a file without implementing any formatting set by the FORMAT, SET, and BREAK commands. The IQF saves the data as it is received from the control tables. Column headings and blank lines are not copied to the file. The DATAONLY option is useful for extracting data from the tables into a file and then loading the data from the file into another application.

To add or replace information in a local file that already exists, specify the REPLACE or APPEND option of the FILE command. Otherwise, IQF sends you an error message and prompts you to indicate REPLACE or APPEND. If you indicate APPEND for a file that does not exist, IQF creates a new file.

If you try to save a query result that is wider than the specified line size, IQF truncates the result to fit on the page and provides a warning message.

Format

```
FILE filename [DATAonly] [{REPlace, APPend}]
```

filename The complete name of a local file that will contain the query result. Include as many as 80 alphanumeric characters. The name must follow the local file-naming conventions.

DATAONLY Specifies that only the data from the query result is copied to the file. No formatting of the data is performed and no blank lines are inserted between the lines of data. Also, no column names are written to the file.

REPLACE Specifies that the contents of the local file should be replaced with the current query result.

APPEND Specifies that the current query result should be appended to the contents of the local file.

Example

```
IQF> SELECT USER_ID, LAST_NAME, USER_GROUP, -
IQF> PROTECTION_GROUP, UPDATE_DATE FROM USERS
IQF> RUN
SELECT USER_ID, LAST_NAME, USER_GROUP, -
PROTECTION_GROUP, UPDATE_DATE FROM USERS
IQF> FILE USER.REPORT
```

The text file `USER.REPORT` is created and contains the following:

```
USER_ID   LAST_NAME  USER_GROUP   PROTECTION_GROUP  UPDATE_DATE
-------   --------   ----------   ----------------  -----------
EDMADMIN  BROWN      ME            F23B               880425
AMELIA    EARHART    EE            E23B               880523
ORVILLE   WRIGHT     EE            DB12               880704
```

To view the contents of the file:

- In IQF, type the following command:

  ```
  IQF> TYPE USER.REPORT
  ```

  OR

- Use the text editor provided with your operating system.

Add the results of the next query to the `USER.REPORT` file as follows:

```
IQF> SELECT UPDATE_TIME, UPDATE_USER_ID FROM USERS
IQF> RUN
SELECT UPDATE_TIME, UPDATE_USER_ID FROM USERS
IQF> FILE USER.REPORT APPEND
IQF> TYPE USER.REPORT

USER_ID   LAST_NAME  USER_GROUP   PROTECTION_GROUP  UPDATE_DATE
-------   --------   ----------   ----------------  -----------
SNOOPY    BROWN      ME            F23B               880425
AMELIA    EARHART    EE            E23B               880523
ORVILLE   WRIGHT     EE            DB12               880704

UPDATE_TIME     UPDATE_USER_ID
-----------     --------------
07:30:45        EDMADMIN
22:15:10        ORVILLE
10:54:17        ORVILLE
16:23:31        AMELIA
01:46:19        EDMADMIN
```

Replace the contents of the `USER.REPORT` file with the results of the next query as follows:

```
IQF> SELECT UPDATE_TIME, UPDATE_USER_ID FROM USERS
IQF> RUN
SELECT UPDATE_TIME, UPDATE_USER_ID FROM USERS
IQF> FILE USER.REPORT REPLACE
IQF> TYPE USER.REPORT

UPDATE_TIME     UPDATE_USER_ID
-----------     --------------
07:30:45        EDMADMIN
22:15:10        ORVILLE
10:54:17        ORVILLE
16:23:31        AMELIA
01:46:19        EDMADMIN
```

Printing the Results of a Query

Use the PRINT command to print the result of the current query. The query result is printed according to the formatting modifications requested or the default values.

The number of characters printed per row and the number of rows printed per page depend on the values you have set previously or the default values. When you try to print a query result that is wider than the specified line size, IQF truncates the result to fit on the page and displays a warning message.

Format

```
PRInt [COPies n]
```

COPIES *n* Indicates that you want more than a single copy; n is a two-digit positive integer that specifies the number of copies you want.

Execute the PRINT command without the COPIES option to print a single copy.

Example

```
IQF> PRINT
```

This command sends a single copy of the current query result to the printer.

```
IQF> PRINT COPIES 6
```

This command sends six copies of the current query result to the printer.

Appendix A

Defining New Views for IQF Use

This appendix explains how to create new views.

- Creating New Views for IQF

Creating New Views for IQF

To create additional views to use with IQF, refer to the book *Vault Database Tables,* which contains information about Vault table and view definitions.

Perform the following steps to create additional views of Vault control tables and make the views available to IQF:

1. Connect to the database as SQL user ID `pdmdm`.

   All views of Vault control tables must be owned by `pdmdm`.
2. Create the required view using the table and column definitions provided in *Vault Database Tables.*
3. Grant the SQL `SELECT` privilege on the created view to SQL user ID `pdmqf`.
4. Create SQL synonyms for the created view so that users will not need to fully qualify the view when using the `SELECT` command.

The SQL commands that create the view `USERS` from the Vault control table `DM_USER` are as shown:

```
CONNECT PDMDM IDENTIFIED BY PDMDM;
DROP VIEW USERS;
CREATE VIEW USERS
(USER_ID,NAME,LAST_NAME,DESCRIPTION,USER_GROUP,
PROTECTION_GROUP,UPDATE_DATE,UPDATE_TIME,UPDATE_USER_ID)
AS SELECT
DM_USER_ID,DM_USER_NAME,DM_USER_LAST_NAME,DM_USER_DESC,
DM_USER_GROUP,DM_USER_PROT_GRP,DM_USER_DATE,DM_USER_TIME,
DM_USER_UPDATER
FROM DM_USER;
GRANT SELECT ON USERS TO PDMQF;
COMMIT WORK;
CONNECT PDMQF IDENTIFIED BY PDMQF;
DROP SYNONYM USERS;
CREATE SYNONYM USERS FOR PDMDM.USERS;
COMMIT WORK;
```

Please note: When you create views, do not use a password column in the view definition. In the view definition shown, the `DM_USER_PW` column containing the user password was omitted from the `USERS` view.

Glossary

ANSI

American National Standards Institute

ASCII character collating sequence

Characters are sorted according to a collating sequence. The ASCII sequence orders numbers 0 to 9 first, then uppercase letters A to Z, and then lowercase letters a to z.

bottom title

The title at the bottom of every page of the query result. Also called a footer.

command file

A local file containing one or more IQF commands.

control table

Information in the database tables that Vault uses to manage data.

current query

The `SELECT` command executed most recently is called the current query.

IQF

Interactive Query Facility

null value

A null value is an empty field and the display shows a blank. When there is no value available to display in the column, the resulting blank in the query result is called a null value.

parameter

A user-defined value that corresponds to a placeholder within a query.

placeholder

A symbol in a `SELECT` command that is replaced with a value that you supply when you execute the query. A placeholder can be either a numeric or an alphabetic string (either numbers or letters) preceded by the ampersand (&) symbol. When IQF encounters a placeholder in a `SELECT` command, it replaces it with the corresponding parameter in the `START` command.

predefined query

IQF is shipped with a number of command files containing queries (`SELECT` commands) and other IQF commands that generate typical reports. *See also* command file.

query result

The report generated by a query.

RDBMS

Relational Database Management System

SQL

Structured Query Language

top title

The title at the top of every page of the query result. Also called a header.

view

A view is an alternate and temporary representation of data from one or more tables. The Vault uses views to allow you to access data in the Vault control table. Using views is one method of safeguarding the integrity of tables.

view definition

This includes the names of the tables and columns that the view refers to for information.

Index